Drowzee: Metamorphic Testing for Fact-Conflicting Hallucination Detection in Large Language Models

NINGKE LI[*], Huazhong University of Science and Technology, China
YUEKANG LI[*], The University of New South Wales, Australia
YI LIU, Nanyang Technological University, Singapore
LING SHI, Nanyang Technological University, Singapore
KAILONG WANG[†], Huazhong University of Science and Technology, China
HAOYU WANG, Huazhong University of Science and Technology, China

Large language models (LLMs) have revolutionized language processing, but face critical challenges with security, privacy, and generating hallucinations — coherent but factually inaccurate outputs. A major issue is fact-conflicting hallucination (FCH), where LLMs produce content contradicting ground truth facts. Addressing FCH is difficult due to two key challenges: **1)** Automatically constructing and updating benchmark datasets is hard, as existing methods rely on manually curated static benchmarks that cannot cover the broad, evolving spectrum of FCH cases. **2)** Validating the reasoning behind LLM outputs is inherently difficult, especially for complex logical relations.

To tackle these challenges, we introduce a novel logic-programming-aided metamorphic testing technique for FCH detection. We develop an extensive and extensible framework that constructs a comprehensive factual knowledge base by crawling sources like Wikipedia, seamlessly integrated into Drowzee. Using logical reasoning rules, we transform and augment this knowledge into a large set of test cases with ground truth answers. We test LLMs on these cases through template-based prompts, requiring them to provide reasoned answers. To validate their reasoning, we propose two semantic-aware oracles that assess the similarity between the semantic structures of the LLM answers and ground truth. Our approach automatically generates useful test cases and identifies hallucinations across six LLMs within nine domains, with hallucination rates ranging from 24.7% to 59.8%. Key findings include LLMs struggling with temporal concepts, out-of-distribution knowledge, and lack of logical reasoning capabilities. The results show that logic-based test cases generated by Drowzee effectively trigger and detect hallucinations. To further mitigate the identified FCHs, we explored model editing techniques, which proved effective on a small scale (with edits to fewer than 1000 knowledge pieces). Our findings emphasize the need for continued community efforts to detect and mitigate model hallucinations.

CCS Concepts: • **Software and its engineering → Software testing and debugging**.

Additional Key Words and Phrases: Large Language Model, Hallucination, Software Testing

[*] Ningke Li and Yuekang Li are co-first authors.
[†] Kailong Wang is the corresponding author.

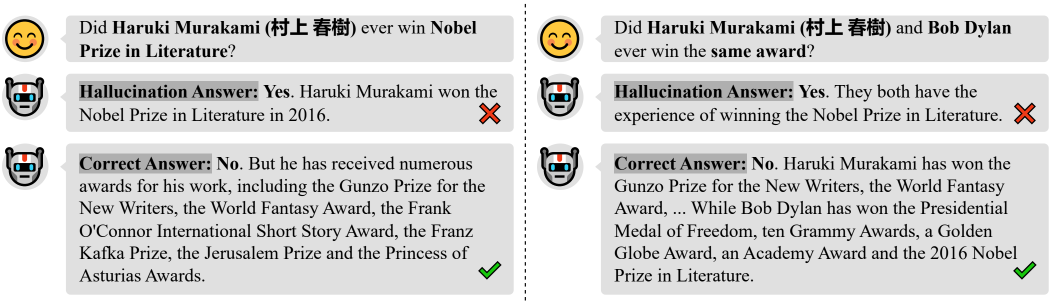
Authors' Contact Information: Ningke Li, Huazhong University of Science and Technology, China, lnk_01@hust.edu.cn; Yuekang Li, The University of New South Wales, Australia, yuekang.li@unsw.edu.au; Yi Liu, Nanyang Technological University, Singapore, yi009@e.ntu.edu.sg; Ling Shi, Nanyang Technological University, Singapore, ling.shi@ntu.edu.sg; Kailong Wang, Huazhong University of Science and Technology, China, wangkl@hust.edu.cn; Haoyu Wang, Huazhong University of Science and Technology, China, haoyuwang@hust.edu.cn.

ACM 2475-1421/2024/10-ART336
https://doi.org/10.1145/3689776

ACM Reference Format:
Ningke Li, Yuekang Li, Yi Liu, Ling Shi, Kailong Wang, and Haoyu Wang. 2024. Drowzee: Metamorphic Testing for Fact-Conflicting Hallucination Detection in Large Language Models. *Proc. ACM Program. Lang.* 8, OOPSLA2, Article 336 (October 2024), 30 pages. https://doi.org/10.1145/3689776

1 Introduction

Large Language Models (LLMs) have brought transformative advancements to the fields of language processing and beyond, showcasing exceptional abilities in text generation and comprehension with wide-ranging applications. However, despite their increasing prevalence, LLMs face critical challenges in security and privacy aspects [Hou et al. 2023; Kaddour et al. 2023; Siddiq and Santos 2023; Xu et al. 2024; Yang et al. 2024a; Zhang et al. 2024], heavily impacting their effectiveness and reliability. A particularly notable issue among these is the phenomenon of "hallucination", where LLMs produce coherent but factually inaccurate or irrelevant outputs during tasks like problem-solving. This tendency to generate misleading information not only jeopardizes the safety of LLM applications but also raises serious usability concerns. Hallucinations in LLMs take several forms, with "Fact-conflicting hallucination" (FCH) being a major concern and the primary type of concern in this paper. FCH is manifested by LLMs generating content that directly contradicts established facts, as exemplified in Figure 1. When an LLM incorrectly believes "Haruki Murakami won the Nobel Prize in Literature in 2016", deviating from the correct answer of "Haruki Murakami has not won the Nobel Prize but other numerous awards for his work in Literature". Such misinformation dissemination leads to significant user confusion, eroding the trust and reliability that are crucial in various LLM applications.



Did **Haruki Murakami (村上 春樹)** ever win **Nobel Prize in Literature**?

Hallucination Answer: **Yes**. Haruki Murakami won the Nobel Prize in Literature in 2016. ✗

Correct Answer: **No**. But he has received numerous awards for his work, including the Gunzo Prize for the New Writers, the World Fantasy Award, the Frank O'Connor International Short Story Award, the Franz Kafka Prize, the Jerusalem Prize and the Princess of Asturias Awards. ✓

(a) Erroneous Knowledge Hallucination

Did **Haruki Murakami (村上 春樹)** and **Bob Dylan** ever win the **same award**?

Hallucination Answer: **Yes**. They both have the experience of winning the Nobel Prize in Literature. ✗

Correct Answer: **No**. Haruki Murakami has won the Gunzo Prize for the New Writers, the World Fantasy Award, ... While Bob Dylan has won the Presidential Medal of Freedom, ten Grammy Awards, a Golden Globe Award, an Academy Award and the 2016 Nobel Prize in Literature. ✓

(b) Erroneous Inference Hallucination

Fig. 1. A Hallucination Output Example.

To address the issue of hallucinations in LLMs, recent studies have introduced a range of methods for their detection and testing. A common and straightforward approach involves creating extensive benchmarks tailored for this purpose. Datasets such as TruthfulQA [Lin et al. 2022], HaluEval [Li et al. 2023a], and KoLA [Yu et al. 2024] have been designed to evaluate hallucinations across different contexts, including question-answering, summarization, and knowledge graphs. Despite the value of these manually labeled datasets, the current techniques for hallucination detection and testing heavily rely on naive and semi-automatic methods, such as string matching, manual validation, or utilizing another LLM for confirmation. This current research landscape in LLM, however, presents a critical gap in automatically and effectively testing FCHs. The main obstacle in testing for FCH is the absence of dedicated ground truth datasets and specific testing frameworks. Unlike other types of hallucinations (e.g., input-conflicting and context-conflicting hallucinations, to be detailed in Section 2.1) which can be identified through checks for semantic consistency, FCH demands the verification of the content's factual accuracy against external sources of knowledge or databases. This requirement makes the process particularly challenging and resource-intensive, especially for tasks processing contents with inherent logical connections.

Bridging the identified research gap in the literature necessitates an exploration of the inherent challenges faced in detecting FCHs, which are crucial for advancing and enhancing the reliability of LLMs. ***Challenge#1: difficulty in automatically constructing and updating benchmark datasets.*** Predominantly, existing methodologies are anchored to manually curated benchmarks. While these benchmarks are effective in detecting certain types of hallucinations, they fall short in encompassing the broad and dynamic spectrum of fact-conflicting scenarios inherent to LLMs. Meanwhile, the need for frequent updates to benchmark data, due to the ever-evolving nature of knowledge, imposes a significant and continuous maintenance effort. The reliance on benchmark datasets thus restricts the detection techniques' adaptability, scalability, and worse, detection capability. ***Challenge#2: difficulty in automatically validating answers from LLM outputs.*** Even when LLMs produce correct final answers, the outputs may not represent the true reasoning process behind them, potentially masking false understanding – a source of FCH hallucination. Automatically validating the reasoning process, especially those involving complex logic relations, is inherently difficult. Furthermore, the consistency in the quality of benchmark questions can vary due to the differing levels of experience and skill among human experts creating them, introducing noise, particularly in data labeling and result validation stages.

Our Work. To address limitations in the existing techniques, we are the first, to the best of our knowledge, to introduce a novel automatic logic-programming-aided metamorphic testing technique for hallucination detection in this work. We have developed an extensive and extensible FCH testing framework, which is based on factual knowledge reasoning and metamorphic testing, seamlessly integrated into Drowzee[1].

Drowzee begins by establishing a comprehensive factual knowledge base, sourced through extensive crawling of information from accessible knowledge bases such as Wikipedia. Each piece of this knowledge acts as a "seed" for subsequent transformations. Leveraging logic reasoning relations, we transform and augment these seeds, thereby expanding the factual knowledge into a well-established set of question-answer pairs. Using the questions and answers in the knowledge set as test cases and ground truth respectively, we construct a reliable and robust FCH testing benchmark. This is implemented through a series of well-formulated template-based prompts to test FCH in LLMs. Specifically, we instruct the LLMs to generate their answers to the test cases. To facilitate a thorough evaluation of the reasoning logic behind their responses, we require the LLMs to provide detailed justifications for their answers. For effective and dependable identification of FCH, we introduce two semantic-aware and similarity-based metamorphic oracles. These oracles operate by extracting essential semantic elements from each sentence and mapping out their logical relationships. By assessing the similarity between the constructed logical and semantic structures of the LLM's answers and the ground truth, we can detect FCH by pinpointing answers that significantly diverge from the ground truth.

Results and Findings. In evaluating our proposed FCH testing framework and Drowzee, we undertake comprehensive experiments to evaluate their effectiveness in a wide array of contexts. On the one hand, our evaluation strategy involves deploying Drowzee across a broad spectrum of topics, sourced from an extensive and diverse range of Wikipedia articles. On the other hand, we examine our framework on a variety of open-source and commercial LLMs, providing a thorough examination of its applicability and performance across different model architectures.

[1] Drowzee is named after a Pokémon [Satoshi Tajiri 2023] character that nourishes itself by eating dreams. This name symbolizes our tool's capability to detect and potentially further assist in eliminating the hallucinations in LLMs.

Our key findings indicate that Drowzee succeeds in automatically generating useful test cases and identifying hallucination issues of six LLMs across nine domains. Using these test sets, we find that hallucination responses generated by different LLMs can vary from 24.7% to 59.8%. We then categorize these hallucination responses into four types. Through an in-depth analysis, we unveil that the lack of logical reasoning capabilities contributes the most to the FCH issues in LLMs. Additionally, we observe that LLMs are particularly prone to generating hallucinations in test cases involving temporal concepts and out-of-distribution knowledge. Furthermore, we confirm that test cases generated using our logical reasoning rules can effectively trigger and detect hallucination issues in LLMs. As mitigation, we investigate the use of model editing techniques [hiyouga 2023; Meng et al. 2022] to rectify the identified FCHs. These techniques have shown promising results when applied on a small scale (involving edits up to less than 1000 pieces of knowledge). Our results highlight the importance of ongoing efforts within the community to detect and address issues of hallucination in LLMs.

Contributions. We summarize the main contributions of this paper below:

- **Development of a novel FCH Testing Framework.** To the best of our knowledge, we are the first to develop a novel testing framework based on logic programming and metamorphic testing to automatically detect FCH issues in LLMs.
- **Construction and Release of Extensive Factual Knowledge Base and Benchmark.** Our work constructs a large-scale benchmark dataset [GitHub 2024] to facilitate collaborative efforts and future advancements in the detection of FCH.
- **Designing and Implementing Innovative Logic-reasoning-based Method for Data Mutation.** We propose and implement five unique logic reasoning rules to mutate and augment the initial seeds from our knowledge base, increasing the diversity and effectiveness of our test scenario.
- **Deployment of FCH-specific semantic-aware testing oracles for automatic LLM answer validation.** We propose and implement two automated verification mechanisms (oracles) that analyze the semantic structure similarity between sentences. These oracles are designed to validate the reasoning logic behind the answers generated by LLMs, hereby reliably detecting the occurrence of FCHs.

2 Background

2.1 Hallucination Categorization

Hallucination in LLMs can be categorized into three main categories [Huang et al. 2023; Yao et al. 2024; Zhang et al. 2023], as detailed below.

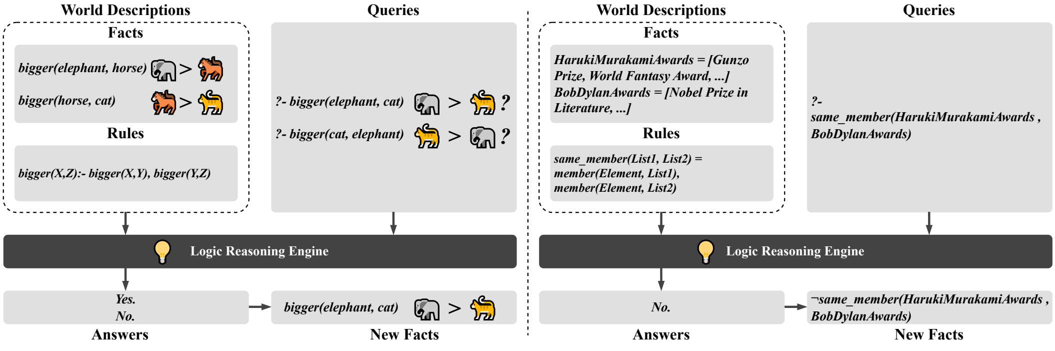
Input-Conflicting Hallucination: This type arises when LLMs produce outputs that are inconsistent with the user's input. This inconsistency can occur in two ways: either the model's response contradicts the task instructions (reflecting a misunderstanding of user intents) or the generated content contradicts the task input (similar to conventional issues in machine translation and summarization). An example of this would be an LLM replacing a key name or detail in a summary, deviating from the actual content provided by the user.

Context-Conflicting Hallucination: In this case, LLMs exhibit contradictions or inconsistencies in lengthy or multi-turn responses. This happens when models lose track of the context or fail to maintain consistency throughout the conversation. Limitations in maintaining long-term memory or identifying relevant context are often the culprits. An instance of context-conflicting hallucination could involve LLMs switching references between two different individuals in a conversation about a specific topic.

Fact-Conflicting Hallucination: This type of hallucination is the key focus of this paper. It occurs when LLMs generate information that is in direct conflict with established world knowledge. This can be due to various factors introduced at different stages of the LLM lifecycle. For example, as shown in Figure 1, an LLM might provide incorrect historical information in response to a user's query, misleading users who are less knowledgeable about the subject.

In this paper, our primary focus is on fact-conflicting hallucinations, a type of error that carries the potential for more serious consequences by misleading users.

2.2 Logic Programming


World Descriptions
Facts
bigger(elephant, horse)
bigger(horse, cat)
Rules
bigger(X,Z):- bigger(X,Y), bigger(Y,Z)
Queries
?- bigger(elephant, cat)
?- bigger(cat, elephant)
Logic Reasoning Engine
Yes.
No.
Answers
bigger(elephant, cat)
New Facts
World Descriptions
Facts
HarukiMurakamiAwards = [Gunzo Prize, World Fantasy Award, ...]
BobDylanAwards = [Nobel Prize in Literature, ...]
Rules
same_member(List1, List2) = member(Element, List1), member(Element, List2)
Queries
?- same_member(HarukiMurakamiAwards , BobDylanAwards)
Logic Reasoning Engine
No.
Answers
¬same_member(HarukiMurakamiAwards , BobDylanAwards)
New Facts

(a) The animal size comparison example

(b) The prize comparison example

Fig. 2. Examples of Logic Programming.

Some existing works [Olausson et al. 2023; Pan et al. 2023; Ye et al. 2023] have already integrated logical programming with large language models in an attempt to enhance their logical reasoning capabilities. In this work, we focus on leveraging logical programming to automate the testing of hallucinations in LLMs. Logic programming languages are declarative, i.e., programming with these languages means describing the world. Using the programs means asking questions about the previously described world. Based on the answers to the questions from the logic reasoning engine, according to the world description, we can derive new facts. Figure 2 shows an example of how logic programming works.

Logic programming allows the programmer to specify the rules and facts, enabling the Prolog interpreter to infer answers to the given queries automatically. Here we explain some key concepts: **Program.** A Prolog program consists of two parts: a list of facts ($\widetilde{R}$) and a list of rules ($\widetilde{Q}$). Throughout the paper, we use the over-tilde notation to denote a list of items. For example, $\widetilde{entity}$ refers to a list of entities, i.e., $entity_1, \ldots, entity_n$.

$$(Program) \quad \mathcal{P} \quad ::= \quad \widetilde{R} +\!+ \widetilde{Q} \tag{1}$$

Facts. A fact is a statement defining a relation as being true. It is made up of a *predicate* and several *entities*. It is denoted as:

$$(Predicate) \quad R \quad ::= \quad nm\,(\widetilde{entity}) \tag{2}$$

An example is $bigger(horse, cat)$, which means horses are bigger than cats. Another example is $member(GunzoPrize, HarukiMurakamiAwards)$, which means that the Gunzon Prize is in the list of prizes awarded to Haruki Murakami.

Rules. A Prolog rule is a Horn clause that comprises a head predicate and a list of body predicates placed on the left and right side of the arrow symbol (:−). A rule means that the left-hand side is logically implied by the right-hand side. The rule bodies are either positive or negative relations, corresponding to the requirements upon the presence or absence of facts. We use "R" and "$\neg R$" as abbreviations for "$\texttt{Pos}\ R$" and "$\texttt{Neg}\ R$", respectively. It is denoted as:

$$\begin{array}{rlcl} (\mathit{Rule}) & Q & ::= & R \text{ :- } \widetilde{body} \\ (\mathit{Rule\ Bodies}) & body & ::= & \texttt{Pos}\ R \mid \texttt{Neg}\ R \end{array} \tag{3}$$

An example is $bigger(X, Z)$:− $bigger(X, Y), bigger(Y, Z)$, which means the $bigger$ relation is **transitive**. Another example is $smaller(X, Y)$:− $bigger(Y, X)$, which means $smaller$ is an **inverse** relation of $bigger$. The last example here is $same\_member(List1, List2)$, which is $true$ if there exists at least one $Element$ that is a member of both $List1$ and $List2$. It is a **composite** type of two $member$ predicates.

Queries. A query has the same structure as the body of a rule, i.e., it is a sequence of predicates separated by commas and executed against a database of facts. The logic reasoning engine will answer Yes if the sequence of predicates in the query is $True$ according to the facts and rules. Otherwise, it will answer No.

An example query is ? - $bigger(elephant, cat)$, which means asking the logic reasoning engine whether elephants are bigger than cats. Another example is ? - $same\_member(HarukiMurakami Awards, BobDylanAwards)$, which means asking if the awards won by Haruki Murakami and Bob Dylan have overlapped.

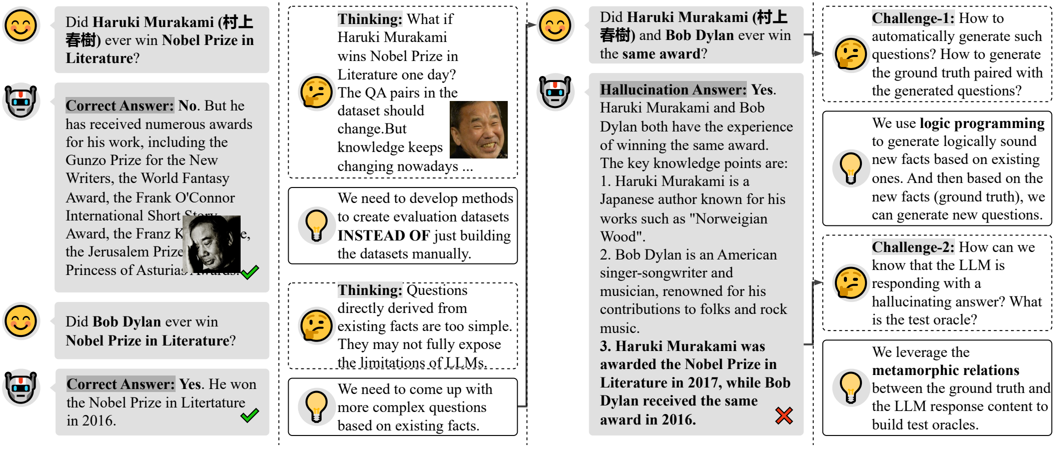
Reasoning Rules. As shown in Figure 2, generating new facts through logic programming requires facts (Equation (2)), rules (Equation (3)), queries, and answers to the queries. To simplify the notation of this process, we bring up the concept of *reasoning rules* in this paper, which describes the inference process of using facts and rules (predicates) to reach the conclusion (a new fact in the form of a predicate) by omitting the process of querying and analyzing the query answers. A reasoning rule is denoted in this form:

$$\frac{R_1, R_2, \ldots}{R_{new}} \tag{4}$$

3 Motivating Example

Figure 3 shows a motivating example of Drowzee. Assume we have the facts about whether Haruki Murakami and Bob Dylan have won the Nobel Prize, as illustrated in the left sub-figure. The question to ask LLMs is straightforward: We can ask whether Haruki Murakami/Bob Dylan has won the Nobel Prize or not. Asking and verifying this knowledge requires no logic reasoning. However, the straightforward questions are often not enough to unveil hallucinations. **Therefore, more diversified questions (questions with intertwined and complex information, as illustrated in the right sub-figure) are needed.**

In order to generate more diversified benchmarks, previous research [Li et al. 2023a; Yu et al. 2024] involves human experts to generate the questions and annotate the answers for hallucination checking. Although the manually generated benchmarks can unveil certain hallucinations, they suffer from several drawbacks. **The landscape of knowledge is dynamic, with new information continuously surfacing and older information becoming obsolete.** If facts change continuously over time, for instance Haruki Murakami were to win the Nobel Prize in the future, this would necessitate regular updates and corrections to the ground truth in existing datasets to reflect them. However, maintaining the accuracy of these benchmarks demands a significant amount of manual labor. Additionally, the quality of the questions might be inconsistent due to the



Did **Haruki Murakami (村上春樹)** ever win **Nobel Prize in Literature**?

Correct Answer: **No**. But he has received numerous awards for his work, including the Gunzo Prize for the New Writers, the World Fantasy Award, the Frank O'Connor International Short Story Award, the Franz K e, the Jerusalem Prize Princess of Asturias Awards. ✓

Did **Bob Dylan** ever win **Nobel Prize in Literature**?

Correct Answer: **Yes**. He won the Nobel Prize in Litertature in 2016. ✓

Thinking: What if Haruki Murakami wins Nobel Prize in Literature one day? The QA pairs in the dataset should change.But knowledge keeps changing nowadays ...

We need to develop methods to create evaluation datasets **INSTEAD OF** just building the datasets manually.

Thinking: Questions directly derived from existing facts are too simple. They may not fully expose the limitations of LLMs.

We need to come up with more complex questions based on existing facts.

Did **Haruki Murakami (村上春樹)** and **Bob Dylan** ever win the **same award**?

Hallucination Answer: **Yes**. Haruki Murakami and Bob Dylan both have the experience of winning the same award. The key knowledge points are:
1. Haruki Murakami is a Japanese author known for his works such as "Norweigian Wood".
2. Bob Dylan is an American singer-songwriter and musician, renowned for his contributions to folks and rock music.
3. Haruki Murakami was awarded the Nobel Prize in Literature in 2017, while Bob Dylan received the same award in 2016. ✗

Challenge-1: How to automatically generate such questions? How to generate the ground truth paired with the generated questions?

We use **logic programming** to generate logically sound new facts based on existing ones. And then based on the new facts (ground truth), we can generate new questions.

Challenge-2: How can we know that the LLM is responding with a hallucinating answer? What is the test oracle?

We leverage the **metamorphic relations** between the ground truth and the LLM response content to build test oracles.

Fig. 3. Motivating Example.

differences in the experience and skills of the human experts who create them. Consequently, the efficiency and soundness of the manually generated benchmarks are not guaranteed.

The limitations of the manually generated benchmarks motivate the need for an automated technique to test for hallucinations in LLMs. Nevertheless, automatically generating diverse benchmarks is challenging. **First, generating suitable and valid questions is challenging (challenge#1).** While it is important for the questions in the testing benchmark to cover a diverse range of scenarios, they cannot be randomly generated or arbitrarily selected. Instead, the questions must be logically coherent and aligned with well-established factual knowledge and ground truth. **Second, deriving the test oracles for detecting hallucinations is challenging (challenge#2).** The LLM's answer is typically expressed in lengthy and potentially complex sentences. The key to determining if an LLM has produced an FCH lies in assessing whether the overall logical reasoning behind its answer is consistent with the established ground truth. Automatically analyzing and comparing the intricate logical structures within the LLM's response and the factual ground truth remains an inherently difficult task.

These two challenges can both be addressed by leveraging logic programming. We can derive new logically sound facts based on existing knowledge. With the new facts, we can then generate diverse questions and their ground truth answers. With the ground truth answers, we can generate test oracles to capture hallucinations. In short, the idea of using logic programming to tackle the challenges motivates the design of Drowzee.

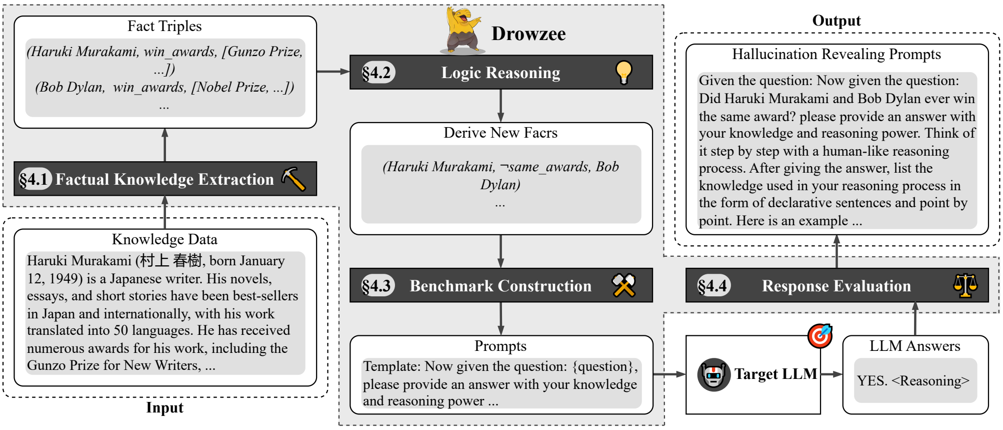
4 Methodology

We design and implement Drowzee to address the aforementioned challenges, the workflow of which is illustrated in Figure 4. Drowzee is comprised of the following four modules, with each module to be detailed later.

- **Factual Knowledge Extraction (§4.1):** Based on voluminous knowledge database dumps, Drowzee acquires fundamental information and factual triples of valid entities.
- **Logical Reasoning (§4.2):** In this module, Drowzee leverages reasoning rules to generate sound and diverse facts as new ground truth knowledge.
- **Benchmark Construction (§4.3):** This module focuses on creating high-quality test case-oracle pairs from the newly-derived ground truth knowledge. The test oracles are generated based

on a simple yet effective metamorphic relation: *Since the newly generated knowledge is sound, the questions complying with the knowledge should be answered with "YES" and the questions contravening the knowledge should be answered with "NO".* This module also includes strategies for effectively and reliably generating or selecting prompts for interaction with LLMs.

- **Response Evaluation (§4.4):** The final module evaluates the responses from the LLMs and detects factual consistency automatically. It first parses LLM outputs using NLP to construct semantic-aware structures, then evaluates their semantic similarity to ground truth. Subsequently, it develops similarity-based oracles applying metamorphic testing to assess consistency between LLM responses and ground truth.


Drowzee
Fact Triples
(Haruki Murakami, win_awards, [Gunzo Prize, ...])
(Bob Dylan, win_awards, [Nobel Prize, ...])
...
§4.1 Factual Knowledge Extraction
§4.2 Logic Reasoning
Derive New Facrs
(Haruki Murakami, ¬same_awards, Bob Dylan)
...
§4.3 Benchmark Construction
Prompts
§4.4 Response Evaluation
Output
Hallucination Revealing Prompts
Knowledge Data
Input
Target LLM
LLM Answers
YES. <Reasoning>

Haruki Murakami (村上 春樹, born January 12, 1949) is a Japanese writer. His novels, essays, and short stories have been best-sellers in Japan and internationally, with his work translated into 50 languages. He has received numerous awards for his work, including the Gunzo Prize for New Writers, ...

Template: Now given the question: {question}, please provide an answer with your knowledge and reasoning power ...

Given the question: Now given the question: Did Haruki Murakami and Bob Dylan ever win the same award? please provide an answer with your knowledge and reasoning power. Think of it step by step with a human-like reasoning process. After giving the answer, list the knowledge used in your reasoning process in the form of declarative sentences and point by point. Here is an example ...

Fig. 4. The Workflow of Drowzee. *The avatar is painted by one of the authors to avoid copyright issues.

4.1 Factual Knowledge Extraction

This step aims to extract fundamental facts from the input knowledge data into fact triples that can be utilized for logical reasoning.

Existing knowledge databases [Auer et al. 2007; Bollacker et al. 2007; Miller 1995; Suchanek et al. 2007] not only encompass a vast array of documents and pages but also provide available structured data. Extracted from knowledge databases, the structured data would become an ideal resource for the construction and enrichment of factual knowledge. Thus, the genesis of our test case data is exclusively rooted in the entities and structured information sourced from current knowledge databases, ensuring a sophisticated and well-informed foundation for our testing framework. Basically, we follow the categorization of entities and relations used by WikiPedia [Auer et al. 2007] to perform the identification. Figure 5a shows the categories of the entities. Figure 5b shows the categories of the relations and some example fact triples.

The detailed process is outlined in Algorithm 1. As defined in the previous Equation 2, we extract the facts in the structure of three-element predicates, i.e., $nm\,(s, o)$, where "s" (stands for *subject*) and "o" (stands for *object*) are entities, and "nm" is the name of the predicate. The facts extraction is done on a per-category basis, implementing a divide-and-conquer strategy, which efficiently integrates all the facts ranging from all the categories. As shown in Algorithm 1, for any given entity category and relation category, the function ExtractGroundFacts iterates through all possible entities and relations. For each combination (*entity*, *nm*), it queries the database using the QueryDB function (Lines 3-6), which retrieves all three-element facts established with the specific predicate *nm* and the argument *entity* placed either in the subject or the object position.

Category Type	Description
Culture and the Arts	Famous films, books, etc.
Geography and Places	Countries, cities and locations.
Health and Fitness	Diseases and disease-causing genes.
History and Events	Famous historical events, etc.
People and Self	Important figures and contributors.
Mathematics and Logic	Common formulas and theorems.
Natural and Physical Sciences	Celestial bodies and astronomy.
Society and Social Sciences	Major social institutions, etc.
Technology and Applied Sciences	Computer science, etc.

(a) Entity Categorization.

Category Type	Example
Noun Phrase	*place_of_birth (Barack Obama, Honolulu).* *genre (28 Days Later, horror film).*
Verb Phrase in Passive Voice	*killed_by (John F. Kennedy, Lee Harvey Oswald).* *located_in_time_zone (Arizona, UTC-07:00).*
Verb Phrase in Active Voice	*follows (4769 Castalia, 4768 Hartley).* *replaces (American Broadcasting Company, NBC Blue Network).*

(b) Relation Categorization.

Fig. 5. Entity and Relation Categorization.

Algorithm 1 Facts Extraction

```
Require: Entity Category ($EC$), Relation Category ($RC$)
Ensure: Ground Facts ($\widetilde{R}_{ground}$)
function ExtractGroundFacts($EC$, $RC$)
    $\widetilde{R}_{ground} \leftarrow []$                    ▷ Initialization
    for $entity \in EC$ do                                   ▷ Iterate over each entity
        for $nm \in RC$ do                                   ▷ Iterate over each relation
            $\widetilde{R} \leftarrow$ QueryDB($entity$, $nm$)     ▷ Retrive ground facts
            $\widetilde{R}_{ground}.append(\widetilde{R})$       ▷ Extend the ground facts
    return $\widetilde{R}_{ground}$                          ▷ Return the ground facts
```

4.2 Logical Reasoning

This step aims to derive additional, enriched information from previously extracted factual knowledge. Drowzee uses a logical programming-based processor to automatically generate new factual knowledge. This allows us to take one or more factual knowledge triples as input and generate a derived triple as output with five types of inference rules.

To tackle the primary concern of generating FCH test cases with variability, we design five types of reasoning rules (Equation (4)) prevalently adopted in several literature [Abboud et al. 2020; Liang et al. 2022; Ren and Leskovec 2020; Tian et al. 2022; Zhou et al. 2019] in the context of knowledge reasoning. This provides sound strategies to prepare new facts for further test case generation. Drowzee will exhaustively apply all the rules to all their relevant fact triples to generate new knowledge. The definitions of the five types of rules are detailed as follows.

Rule#1: Negation Reasoning. Based on a given factual knowledge, we can determine whether the opposite of this fact is correct or incorrect by applying Definition 1.

Definition 1. ***Negation Reasoning Rule*** $[Neg]$. *Given a factual knowledge triple* (s, nm, o)*, then we can derive the new triple* $(s, \overline{nm}, o)$ *is not valid.* $\overline{nm}$ *indicates the negation of the relation nm.*

$$\frac{nm(s,o)}{\neg\ \overline{nm}(s,o)}[Neg]$$

An example of this type of rule is: $\dfrac{was(s,o)}{\neg\ wasn't(s,o)}[Neg]$.

With this rule, from the triple *(Haruki Murakami, won, the Nobel Prize in Literature in 2016)*, we derive that the negation of this triple *(Haruki Murakami, did not win, the Nobel Prize in Literature in 2016)* contains false factual knowledge.
Rule#2: Symmetric Reasoning. In symmetric relations, if the subject and object in a triple maintain coherence upon interchange, a new triple can be deduced in accordance with Definition 2.

Definition 2. ***Symmetric Reasoning Rule*** $[Sym]$. *Given a factual knowledge triple* (s, nm, o), *then we can derive a new triple* (o, nm, s).

$$\frac{nm(s,o)}{nm(o,s)}\,[Sym]$$

An example of this type of rule is: $\dfrac{different\_from(s,o)}{different\_from(o,s)}\,[Sym]$.
With this rule, from the original triple *(Haruki Murakami, different_from, Haruki Uemura)*, we derive a new triple *(Haruki Uemura, different_from, Haruki Murakami)* (Haruki Uemura is a Japanese judoka). Note that the symmetric reasoning rule is primarily utilized within the composition reasoning rule (to be detailed next) and does not introduce new knowledge on its own.
Rule#3: Inverse Reasoning. In an inverse relation, the subject and object can be reversely linked through a variant of the original relation, as defined in Definition 3.

Definition 3. ***Inverse Reasoning Rule*** $[Inverse]$. *Given a factual knowledge triple* (s, nm, o) *and a reversed relation* nm' *of R, then we can derive a new triple* (o, nm', s).

$$\frac{nm(s,o),\, nm' = Reverse(nm)}{nm'(o,s)}\,[Inverse]$$

An example of this type of rule is: $\dfrac{influence\_by(s,o)}{influence(o,s)}\,[Inverse]$. With this rule, from the triple *(Haruki Murakami, influence_by, Richard Brautigan)*, we can derive a new triple *(Richard Brautigan, influence, Haruki Murakami)*.
Rule#4: Transitive Reasoning. In transitive relations, if the object in one triple is the subject of the second triple, we can therefore derive a new triple following the Definition 4.

Definition 4. ***Transitive Reasoning Rules*** $[Trans]$. *Given two factual knowledge triples* (s_1, nm, o_1) *and* (s_2, nm, o_2), *if* o_1 *is semantically equivalent to* s_2, *then we can derive a new triple* (s_1, nm, o_2).

$$\frac{nm(s_1,o_1),\ nm(s_2,o_2),\ o_1 = s_2}{nm(s_1,o_2)}\,[Trans]$$

An example here is:

$$\frac{loc\_in(s_1,o_1),\ loc\_in(s_2,o_2),\ o_1 = s_2}{loc\_in(s_1,o_2)}\,[Trans].$$

With this rule, from triples *(Haruki Murakami, locate_in, Kyoto)* and *(Kyoto, locate_in, Japan)*, we derive a new triple *(Haruki Murakami, locate_in, Japan)*.
Rule#5: Composite Reasoning. The previous four reasoning rules are all meta-rules capturing the most basic and fundamental logical relations among the facts and rules. Several basic reasoning rules can be chained together to form a composition reasoning rule if the relations in the rules

have logical relations. Composite reasoning rules can generate knowledge that requires multiple steps of reasoning.

Definition 5. ***Composite Reasoning Rules*** $[Comp]$**.** *Given multiple basic reasoning rules or predicates* $[Rule_i] \in \{[Neg], [Sym], [Inverse], [Trans], [Predicates]\}$, *we can chain them up to form a new composite reasoning rule.*

$$\frac{\begin{array}{c}\dfrac{nm_{1\_Rule_1}(\ldots), nm_{2\_Rule_1}(\ldots), \ldots}{R_1}\,[Rule_1], \ldots \\ \hline \dfrac{\ldots}{\ldots}\,[\ldots], \ldots \\ \hline \dfrac{nm_{1\_Rule_i}(\ldots), nm_{2\_Rule_i}(\ldots), \ldots}{R_i}\,[Rule_i], \ldots\end{array}}{R_{new}}\,[Comp]$$

The process of applying these various rules to the ground truth triples extracted in the previous module can be referenced in Algorithm 2. An automatic rule generator could be designed at the first stage to iterate its predicates and generate the derivation rules Q according to the relation category (as in Lines 3-4). The corresponding query problems are also generated and mapped to the generated rules, which could be applied to the Prolog query later. With the predetermined rules, we can be assisted with the Prolog engine, asserting all the related triples and consulting the reasoning rules, as outlined in Lines 5-6. We use $[\![R]\!]_{\mathcal{P}}$ to denote the query results of R w.r.t the Prolog program $\mathcal{P}$. When R contains no variables, it returns Boolean results indicating the presence of the fact; otherwise, it outputs all the possible instantiations of the variables. Then as stated in Lines 7-9, by obtaining solutions from Prolog, we can generate new knowledge triples based on the entities and their relations provided. For each instantiation that contains one subject "s" and one object "o", we then compose them with the new predicate, which is taken as one derived fact. These derived facts are later used to generate test cases.

Algorithm 2 Deriving New Facts

```
Require: Ground Facts ($\widetilde{R}_{ground}$), Relation Category ($RC$)
Ensure: Derived Facts ($\widetilde{R}_{derived}$)
 1: function DerivingFacts($\widetilde{R}_{ground}$, $RC$)
 2:     $\widetilde{R}_{derived} \leftarrow []$                          ▹ Initialization
 3:     for $nm$ in $RC$ do                               ▹ Iterate each predicate
 4:         $nm \hookrightarrow (nm_{new}, Q)$      ▹ Obtain the reasoning rule, and the new predicate
 5:         $\mathcal{P} \leftarrow \widetilde{R}_{ground}{+}{+}Q$             ▹ Construct the Prolog program
 6:         $instantiations \leftarrow [\![nm_{new}(S, O)]\!]_{\mathcal{P}}$    ▹ Obtain concrete entities
 7:         for (s, o) in $instantiations$ do                ▹ Iterate each entity tuple
 8:             $R_{new} \leftarrow nm_{new}(s, o)$                ▹ Construct the derived fact
 9:             $\widetilde{R}_{derived}.append(R_{new})$          ▹ Append the derived facts
10:     return $\widetilde{R}_{derived}$                          ▹ Return the derived facts
```

4.3 Benchmark Construction

From the derived triples, this module outlines our approach to constructing question-answer (Q&A) pairs and prompts to facilitate the automatic testing of FCH.

In addressing the obstacle of high human effort demanded in the test oracle generation process, we design an automated generation of test case-oracle pairs based on mapping relations between various entities to problem templates, greatly reducing reliance on manual effort.

Question Generation. To ensure effective and systematic test cases and prompt generation, we have adopted a method that utilizes entity relations mapping to predefined Q&A templates. In the construction of relation-based Q&A templates, one key aspect lies in aligning various types of relations with the corresponding question templates from the derived triples, i.e., the predicate type in the triple. Different relation types possess unique characteristics and expressive requirements, leading to various predefined templates. As listed in Table 1, we map the relation types to question templates based on speech and the grammatical tense of the predicate, to guarantee comprehensive coverage. Beyond these universal templates, for hard-to-describe predicates, we employ customized templates to generate valid Q&A pairs. To enhance the construction of natural language formatted questions, we also leverage the LLM to refine the structure of Q&A pairs.

Another key aspect is regarding the automatic and reliable answer generation. We note that the answer to the corresponding question is readily attainable from the factual knowledge in the triple. Primarily, it is easy to determine whether the answer is true/false based on the derived triples. Meanwhile, mutated templates with positive and negative semantics via the usage of synonyms or antonyms, which greatly enhance the question diversity, can be treated in a similar manner as the negation rule defined in Section 4.2. Specifically, if the answer to a question with original semantics is Yes/No, then for a question with mutated opposite semantics, the corresponding answer would naturally be the opposite, i.e., No/Yes. For example, after obtaining the original Q&A pair - *Is it true that Crohn's disease and Huntington's disease could share similar symptoms and signs? - Yes.*, we can use some antonyms to mutate it into - *Is it true that Crohn's disease and Huntington's disease have totally different symptoms and signs? - No.*

Table 1. Relation-Template Mapping Pattern.

Relation	**Template Examples**
Noun Phrase	- Is it true that ⟨*Subject*⟩ and ⟨*Object*⟩ share ⟨*Relation*⟩? - ⟨*Subject*⟩ and ⟨*Object*⟩ have/made/shared totally different ⟨*Relation*⟩. Please judge the truthfulness of this statement.
Verb Phrase (Passive Voice)	- Is it true that ⟨*Subject*⟩ is/was/are/were ⟨*Relation*⟩ ⟨*Object*⟩? - It is impossible for ⟨*Subject*⟩ to be ⟨*Relation*⟩ ⟨*Object*⟩. Am I right?
Verb Phrase (Active Voice)	- Is it true that ⟨*Subject*⟩ ⟨*Relation*⟩ ⟨*Object*⟩? - ⟨*Subject*⟩ ⟨*Relation*⟩ ⟨*Object*⟩.

Table 2. Prompt Template.

Instruction: Answer the question with your knowledge and reasoning power.
Query: Now given the question: *question*, please provide an answer with your knowledge and reasoning power. Think of it step by step with a human-like reasoning process. After giving the answer, list the knowledge used in your reasoning process in the form of declarative sentences and point by point. Here is an example. Question: During Barack Obama held the position as the president of the USA, were any films directed by James Cameron released? Supposed Response: Yes, during Barack Obama's presidency from 2009 to 2017, one film directed by James Cameron was released - Avatar in 2009. The key knowledge points used in this reasoning process are: 1. Barack Obama was the US President from January 20, 2009 to January 20, 2017. 2. James Cameron is a famous film director known for movies like Titanic, Avatar, Terminator 2, etc. 3. Cameron's only film release during Obama's presidency was Avatar in 2009.

Prompt Construction. As illustrated in Table 2, before initiating our interaction with LLMs, we predefine specific instructions and prompts, requesting the model to utilize its inherent knowledge and inferential capabilities to deliver explicit (yes/no/I don't know) judgments on our queries. Additionally, we instruct the model to present its reasoning process in a template following the judgment. The primary aim is to ensure LLMs provide easily assessable responses by using standardized prompts and instructions. This approach also ensures that the model can exercise its reasoning abilities as effectively as possible under the given instructions and cues.

4.4 Response Evaluation

The objective of our proposed module is to enhance the detection of FCH within LLM outputs, specifically focusing on the discrepancies between LLM responses and verified ground truth in Q&A pairs. Recognizing the inherent challenges in directly accepting "yes" or "no" answers from LLMs due to potential inaccuracies, our approach underscores the importance of thoroughly analyzing the reasoning process presented by LLMs. This analysis is vital for accurately determining the factual consistency of LLM responses, thereby addressing the primary challenge in identifying FCH within LLM outputs.

To achieve automated detection of factual consistency, our methodology first incorporates a parsing step that leverages advanced NLP techniques. This step is designed to extract essential semantic elements from each sentence within LLM outputs, assembling these elements into a coherent, semantic-aware structure. The foundational premise of our approach is predicated on evaluating the semantic similarity between these constructed structures, aiming to discern the degree of consistency in their underlying semantics. Subsequently, we propose the development of a list of similarity-based testing oracles. These oracles are instrumental in applying metamorphic testing principles, enabling us to systematically assess the consistency or inconsistency between LLM responses and the established ground truth. Note that our focus is on the accuracy of ground truth facts rather than highly specialized or sequential content like mathematical proofs. Consequently, during evaluation, we emphasize whether the entities and relations in the response align with the ground truth, regardless of the order in which the facts are presented. Our approach is structured around several critical steps, detailed as follows:

Step 1. Preliminary Screening. First, we eliminate scenarios in which the LLM declines to provide an answer, as indicated by the "answer" field of the LLM's response (as described in Algorithm 3 Lines 3-4). Most of these responses arise because the LLM lacks the relevant knowledge for the reasoning process. Since these responses adhere to the LLM's principle of honesty, we classify them as correct and normal responses, denoted as *CO* in the algorithm.

Step 2. Response Parsing and Semantic Structure Construction. As stated in Algorithm 3 Lines 6-7, for the remaining suspicious responses, the ExtractTriple function is used to generate triples based on the statements contained in the *reasoning process* part of the LLM's response. Then from the extracted triples ($\widetilde{Trpl}$), the BuildGraph function can construct a semantic structure SS_{resp}, where the *entities* (i.e., the subject and object) are represented as nodes, and the *relation* between them is illustrated as an edge connecting these nodes. Concurrently, the ground truth triples ($\widetilde{R_{all}}$) associated with the question are used as input to construct a similar semantic structure SS_{GT}.

Step 3. Similarity-based Metamorphic Testing and Oracles. We apply metamorphic relations to detect and evaluate potential errors in LLM responses, based on the relationships between inputs and outputs, rather than relying on traditional labeled data. In our context, metamorphic relations specifically refer to comparing the similarity between semantic structures generated by LLMs and the ground truth counterparts, to identify and classify hallucination answers from LLMs (as mentioned in Algorithm 3 Lines 8-17). Note that we provide four classifications: correct responses (denoted as *CO*), hallucinations caused by error inference (*EI*), hallucinations caused by erroneous knowledge (*EK*), and hallucinations containing both issues (*OL*).

Algorithm 3 Response Evaluation

```
Require: LLM Response ($Resp$), All Ground Facts ($\widetilde{R}_{all}$), Threshold ($\theta_e, \theta_n$)
Ensure: Evaluation Result Category ($CO, EK, EI, OL$)
function EvaluateResponse($Resp$, $\widetilde{R}_{all}$, $\theta_e$, $\theta_n$)
    $CO, EK, EI, OL \leftarrow []$                                              ▹ Initialization
    if $Resp.answer = refusal$ then
        $CO.append(Resp)$                                                    ▹ Preliminary Screening
    else
        $\widetilde{Trpl} \leftarrow$ ExtractTriple($Resp.reasoning$)                          ▹ Extract useful triples
        $\widetilde{SS}_{resp}, \widetilde{SS}_{ground} \leftarrow$ BuildGraph($\widetilde{Trpl}, \widetilde{R}_{all}$)      ▹ Build semantic structure
        $s_e \leftarrow \mathcal{J}\_Sim_e(\widetilde{SS}_{resp}, \widetilde{SS}_{ground})$                      ▹ Calculate edge similarity
        $s_n \leftarrow \mathcal{J}\_Sim_n(\widetilde{SS}_{resp}, \widetilde{SS}_{ground})$                      ▹ Calculate node similarity
        if $s_e < \theta_e$ and $s_n < \theta_n$ then
            $OL.append(Resp)$                           ▹ Append hallucination related to both types
        else if $s_e < \theta_e$ then
            $EI.append(Resp)$                                  ▹ Append error inference hallucination
        else if $s_n < \theta_n$ then
            $EK.append(Resp)$                                 ▹ Append error knowledge hallucination
        else
            $CO.append(Resp)$                                           ▹ Append correct response
    return $CO, EK, EI, OL$                             ▹ Return the categorized evaluation result
```

Specifically, the oracles for metamorphic testing can be divided into the following types:

Edge Vector Metamorphic Oracle (*MO_E*): This oracle is based on the similarity of edge vectors between SS_{resp} and SS_{ground}. If the vector similarity between the edges in the SS_{resp} and those in SS_{ground} falls below a predetermined threshold, it indicates that the LLM's answer significantly diverges from the ground truth, suggesting the presence of an FCH. Conversely, if the similarity meets or exceeds the threshold, the LLM's answer is considered to align with the ground truth. More specifically, we utilize Jaccard Similarity [ScienceDirect 2023] to gauge the similarity score between edge vectors extracted from SS_{resp} and those in SS_{ground}.

$$\mathcal{J}\_Sim_E(SS_{resp}, SS_{ground}) = \frac{|\widetilde{E}_{resp} \cap \widetilde{E}_{ground}|}{|\widetilde{E}_{resp} \cup \widetilde{E}_{ground}|},$$

check if

$$\mathcal{J}\_Sim_E(SS_{resp}, SS_{ground}) < \theta_n$$

where $\widetilde{E}_{resp}$ and $\widetilde{E}_{ground}$ denote the list of edges extracted from SS_{resp} and SS_{ground}, and θ_E is a predefined threshold (to be detailed in Section 5.1). Intuitively, the similarity score is calculated as the proportion of identical edges shared between the two lists against the total number of unique edges in both lists. If the score is smaller than the threshold, then an FCH is detected. Note that when determining the joint and union of lists $\widetilde{E}_{resp}$ and $\widetilde{E}_{ground}$, we consider two edges as identical if their corresponding relations are identical or represented by synonymous words, and vice versa.

Node Vector Metamorphic Oracle (*MO_N*): This relation examines the similarity of node vectors between SS_{resp} and SS_{ground}. Defined in a similar manner as MO_N, if the node similarity between the edges in the SS_{resp} and those in SS_{ground} falls below a predetermined threshold, it indicates that the LLM's answer significantly diverges from the ground truth, suggesting the presence of an FCH; vice versa. MO_N can be captured by the Jaccard Similarity, defined as follows:

$$J\_Sim_N(SS_{resp}, SS_{ground}) = \frac{|\widetilde{N}_{resp} \cap \widetilde{N}_{ground}|}{|\widetilde{N}_{resp} \cup \widetilde{N}_{ground}|},$$

check if

$$J\_Sim_N(SS_{resp}, SS_{ground}) < \theta_n$$

where $\widetilde{N}_{resp}$ and $\widetilde{N}_{ground}$ denotes the list of nodes extracted from SS_{resp} and SS_{ground}, and θ_n is a predefined threshold (to be detailed in Section 5.1). Intuitively, the similarity score is calculated as the proportion of identical nodes shared between the two lists against the total number of unique nodes in both lists. If the score is smaller than the threshold, then an FCH is detected. Note that when determining the joint and union of lists $\widetilde{N}_{resp}$ and $\widetilde{N}_{ground}$, we consider two nodes as identical if their corresponding entities are identical or represented by synonymous words, and vice versa.

5 Evaluation

Our evaluation targets the following research questions:

- **RQ1 (Effectiveness): How effective is Drowzee for identifying LLM FCH issues?** This RQ studies the effectiveness of Drowzee in generating test cases and identifying LLM FCH issues.
- **RQ2 (Hallucination Categorization and Analysis): What is the categorization of LLM FCH issues?** This RQ categorizes the FCH issues of various LLMs identified by Drowzee. We also provide case studies for some specific cases, including temporal-related FCHs and out-of-distribution-data knowledge-related FCHs.
- **RQ3 (Comparison with Existing Works): How does Drowzee compare with existing approaches in detecting LLM FCH issues?** This RQ investigates whether Drowzee outperforms existing testing benchmarks and methods in constructing test cases and identifying LLM FCH issues. We conduct a qualitative analysis as well as a small-scale quantitative analysis of the accuracy of current hallucination detection methods compared with Drowzee.
- **RQ4 (Ablation Study): Whether the four types of logic reasoning rules can identify LLM FCH issues independently?** This RQ explores whether the logic reasoning rules of Drowzee can effectively identify LLM FCH issues separately.

5.1 Experimental Setup

Knowledge Extraction. We use Wikipedia and Wikidata as sources to extract entities and structured information as base factual knowledge. After downloading the latest Wikipedia dump, we employ wikiextractor [Attardi 2015] to extract relevant text from Wiki pages. In parallel, we invoke Wikidata's SPARQL [Prud'hommeaux and Seaborne 2018] query module for the extraction of triples. Through data processing involving simplification and filtration, we amass a collection of basic factual knowledge, encompassing a sizeable number of 54,483 entities and 1,647,206 triples.
Logic Reasoning Processor. For the logic reasoning module, we apply SWI-Prolog [Wielemaker et al. 2008], an open-source advanced logical programming interpreter. To effectively prevent errors due to excessive stacked strings, and ensure the proper operation of the logical processor when inserting a large number of facts into Prolog, we employ a sampling method and extract a subset of entities to form a query.
Models Under Test. To guarantee a reliable evaluation for RQ1 and RQ2, we evaluate six state-of-the-art large language models with Drowzee. Considering the diverse nature of LLMs, we select two distinct categories for in-depth analysis: the first category comprises API-accessible models with closed-source architecture including ChatGPT (gpt-3.5-turbo-0613) and GPT-4 [OpenAI 2023],

and the second category consists of open-source LLMs with deployability, including Llama2-7B-chat-hf, Llama2-70B-chat-hf [Touvron et al. 2023], Mistral-7B-Instruct-v0.2 [Jiang et al. 2023], and Mixtral-8x7B-Instruct-v0.1 [Jiang et al. 2024].

Model Configuration. We set the *temperature* parameter to 0 to achieve more stable and conservative model outputs, ensuring consistency in the content generated during LLM testing. Additionally, we set the *top-p* value to 0.9 and disable *top-k* (set to 0) to filter out low-probability tokens and select the most likely tokens, thereby improving the accuracy of the generated results. To further validate the consistency of the LLM responses, we conducted significance tests to validate the consistency of the LLM responses. Specifically, we randomly selected 40 test cases from each domain under each rule, resulting in a total of 1440 test cases. Using GPT-3.5-turbo as an example, each test case is run five times under the previously described configuration to interact with the LLM. We then use the Sentence-bert model [Reimers and Gurevych 2019] to calculate the pairwise cosine similarity between the five LLM responses generated for each test case. The consistency of the LLM responses is evaluated using Friedman tests [ScienceDirect 2024], a non-parametric statistical test commonly employed to detect differences in treatments across multiple test attempts. The results show no significant differences between the responses of different runs, with an approximate average p-value of 0.54. This confirms that the generated responses are consistent across multiple executions, justifying the use of a single run for evaluation purposes.

Response Validation Threshold θ. To validate responses from LLMs as described in Section 4.4, we apply StanfordOpenIE [Angeli et al. 2015; Remy 2020] for knowledge triple extraction from LLM responses and then use Phrase-BERT [Wang et al. 2021] to calculate the vector similarity of nodes and edges from the constructed semantic structures. We also utilize GPT-4 to extract triples for some complex responses that StanfordOpenIE cannot handle effectively. Here we set the threshold to 0.8, considering knowledge triples as semantically equivalent if they exceed this threshold, and vice versa. To determine the threshold value, we sample 30 test cases and corresponding LLM responses from each of the nine knowledge domains listed in Figure 5a. Through this analysis, we find that by setting the threshold values for both θ_E and θ_N at 0.8, with the given 270 test cases that are correctly classified, we can estimate the true positives among all test cases through *Laplace's approach in the Sunrise problem* [Laplace 1951], resulting in 99.6% when detecting non-equivalent LLM answers as FCHs. In other words, all instances where an LLM's answer has a semantic similarity score below 0.8 compared to the ground truth were correctly identified as FCH cases.

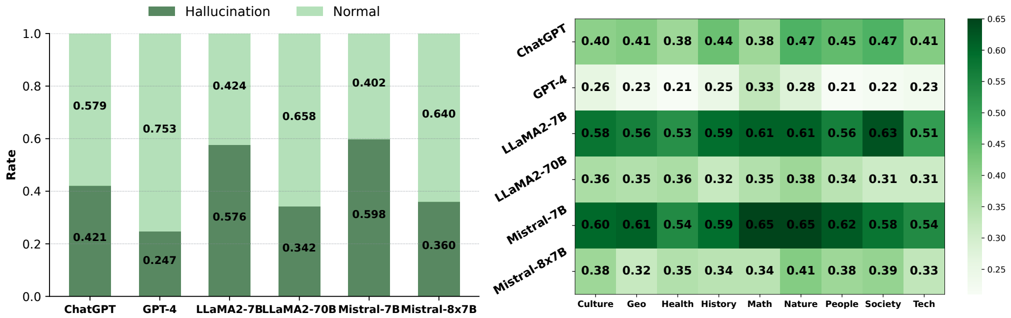
Consistency of Words. To ensure word consistency in experiments, we maintain several dictionaries. For symmetric relations, we have a dictionary that includes relations which retain their meaning when the subject and object are reversed. Additionally, we use synonym dictionaries provided by NLP libraries (e.g., WordNet [Miller 1995]) along with our own set of synonyms tailored for specific cases when validating the LLM responses.

Running Environment. Our experiments are conducted on a server running Ubuntu 22.04 with two 64-core AMD EPYC 7713, 512 GB RAM, and two NVIDIA A100 PCIe 80GB GPUs. Our experiments consume a total of 120 GPU hours.

5.2 RQ1: Effectiveness

To reveal the effectiveness of Drowzee, we evaluate the statistics of test cases generated by Drowzee and then evaluate the capabilities of identifying LLM FCH issues with the generated cases. To further assess the effectiveness of test cases for uncovering FCH issues in specific knowledge domains, we evaluate the performances of LLMs on test cases across various knowledge domains.

Effectiveness on Generating Q&A Test Cases. We apply Drowzee to generate a Q&A test benchmark, amounting to a comprehensive total of 7,200 test cases, designed to provide a broad and detailed evaluation of LLM FCH issues across specific knowledge domains.



(a) Overall Hallucination Rate of Various LLMs. (b) Hallucination Rate Heatmap of Specific Domain.

Fig. 6. Effectivess of Drowzee.

Effectiveness across LLMs. We instruct LLMs under test utilizing Q&A pairs derived from Drowzee and automatically label both hallucination and normal responses. Different LLMs might trigger different hallucinations on the same questions. The results are presented in Figure 6a, illustrating the proportion of FCHs versus normal responses from LLMs under test.

Among all models, GPT-4 exhibits the best performance, demonstrating the lowest proportion of hallucinatory responses in the test cases generated by Drowzee, at only 24.7%, while ChatGPT falls slightly behind with 42.1%. Open-source LLMs including Llama2-7B-chat-hf and Mistral-7B-Instruct-v0.2 with fewer parameters perform worse, but their counterparts with larger parameters (i.e., Llama2-70B-chat-hf and Mixtral-8x7B-Instruct-v0.1) achieve higher normal response rates surpassing ChatGPT on Drowzee. This indicates that the test cases generated by Drowzee successfully trigger hallucination responses across various LLMs when confronted with questions requiring logical reasoning capabilities.

Effectiveness on Specific Domain Knowledge for Each LLM. To further explore the effectiveness of Drowzee in identifying FCH issues spanning various domains of LLMs, we compare hallucination response across nine specific domain knowledge. Figure 6b presents the generated heatmaps of the confusion matrices for hallucination response rate from the specific knowledge field based on the testing results. It can be clearly observed that different models exhibit varying strengths and weaknesses across distinct knowledge domains.

An interesting finding is that, within the domains of natural sciences and mathematics, LLMs generally exhibit weaker performance. This is potentially because there are many astrophysical or mathematical entities and their interrelationships in generated test cases by Drowzee. To answer such questions, the LLM needs an extensive understanding of astrophysical knowledge and mathematical theory. Thus, we infer that this realm of knowledge is not well-covered in the training datasets of LLMs under test, thereby resulting in high hallucination rates. Such a disparity in knowledge is likely a significant factor in the observed underperformance of LLMs in these specific domains.

ANSWER to RQ1

Our evaluation using Drowzee reveals that existing LLMs have a notable tendency to produce FCH when faced with logical reasoning challenges. The results varied across knowledge domains, highlighting that LLMs are more prone to FCH when answering questions that require highly specialized, domain-specific knowledge.

5.3 RQ2: FCH Categorization and Analysis

5.3.1 FCH Categorization. We categorize the hallucination responses in more detail and focus primarily on two types of hallucination: error knowledge response and error inference response. Note that we consider refusal to respond such as 'I don't know' due to the lack of relevant knowledge as adhering to LLMs' honesty and truthfulness principles. Therefore, we categorize refusal to respond as a normal response. To ensure fair and unbiased categorization, 100 hallucination-related responses were randomly selected and independently categorized by three co-authors, who then discussed the results to reach a consensus categorization.

Error Knowledge Response. Originated from LLMs utilizing erroneous or contextually inappropriate knowledge during the reasoning process.

Error Inference Response. The most frequently occurring type is attributed to the lack of reasoning power and flawed reasoning thoughts of LLMs.

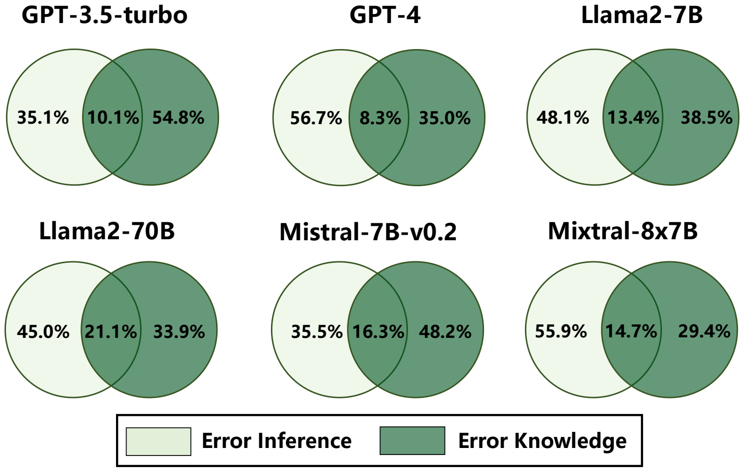
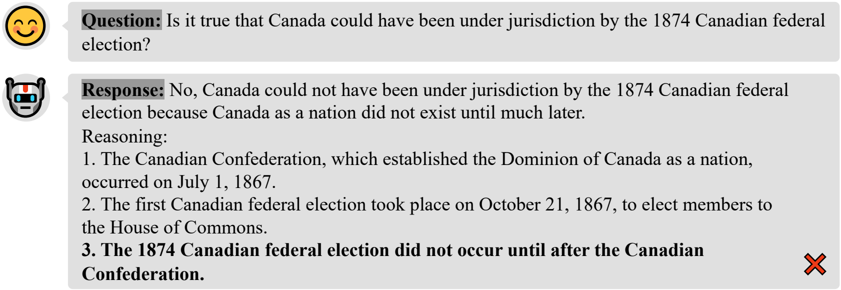


Fig. 7. FCH Categorization.

5.3.2 Hallucination Measurement. Here we provide the distribution of the hallucination categorization results, as demonstrated in Figure 7. There is partial overlap between these two types of hallucinations because incorrect reasoning processes may also involve erroneous knowledge. Among these issues, there are several contradictory answers primarily arising from inconsistency between incorrect reasoning processes and correct answers; thus, it exists in these two types of errors. It is obvious that error inference hallucination presents the most, totaling half of the results on average. This indicates that the primary cause of FCH issues in logical reasoning is the insufficient reasoning capability of LLMs. Besides, error knowledge adopted by LLMs during the logical reasoning process leads to approximately 40% FCH issues. The overlaps account for about 8%-21% at the hallucination ratio, which indicates there are entities where LLMs have learned entirely erroneous relevant information, necessitating the employment of measures for correction.

5.3.3 Case Study. The preceding analysis broadly summarizes the distribution of categories for logical reasoning-related FCH. According to our investigation, error inference response and error knowledge response are the most prevalent two types.

Error Inference Hallucination. One of the most common types of logical reasoning leading to error inference hallucination is temporal attribute reasoning, proven to be a category of reasoning



Question: Is it true that Canada could have been under jurisdiction by the 1874 Canadian federal election?

Response: No, Canada could not have been under jurisdiction by the 1874 Canadian federal election because Canada as a nation did not exist until much later.
Reasoning:
1. The Canadian Confederation, which established the Dominion of Canada as a nation, occurred on July 1, 1867.
2. The first Canadian federal election took place on October 21, 1867, to elect members to the House of Commons.
3. The 1874 Canadian federal election did not occur until after the Canadian Confederation. ✗

Fig. 8. Error Inference Hallucination Example from Mistral-7B-v0.2.

task that performs poorly on LLMs [Qiu et al. 2024]. Experiments on time-related reasoning tasks are comprehensively conducted and unsatisfactory performance of LLMs are observed.

As illustrated in Figure 8, error inference with correct knowledge leads to a hallucination response from Mistral-7B-v0.2. As knowledge provided by the LLM reasoning process, it is clear that the answer should be 'Yes' as the 1874 Canadian federal election applies to the jurisdiction of Canada. However, it appears that the LLM has become ensnared by its limitations.

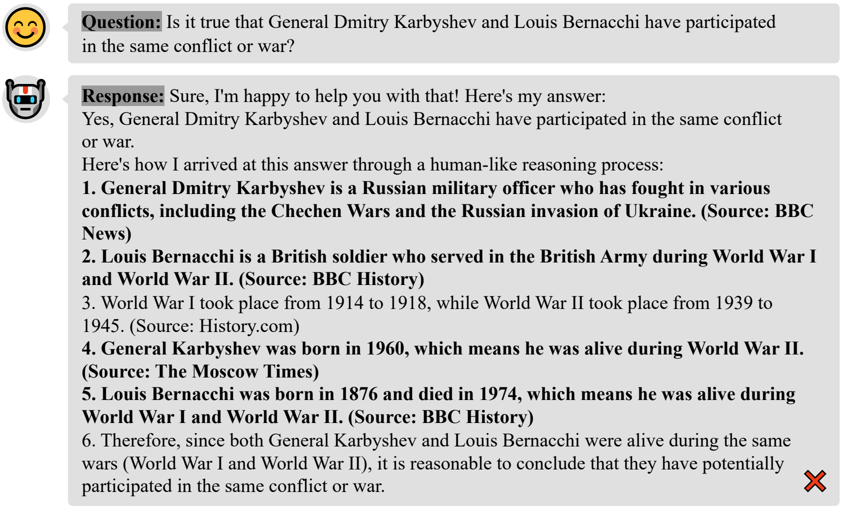
A possible explanation for this phenomenon is that the LLM does not utilize its reasoning abilities but rather relies on unreliable intuition to respond when faced with a question lacking detailed instructions. This insight inspires us to explore methods for effectively enhancing the reasoning capabilities of LLMs through a single interaction, guiding these models toward uncovering answers in a way that mirrors human reasoning processes.

Finding 1. LLMs exhibit weaker performance in sensitivity to temporal information, as well as in their ability to discern sequential logic, which may result in error inference hallucination.

Error Knowledge Hallucination. Figure 9 demonstrates a classic example of LLM hallucination caused by using error knowledge for logical reasoning. General Dmitry Karbyshev (1880-1945) was a Russian Imperial Army soldier who served in several wars during World War I (1914-1918) and II (1939-1945), and Louis Bernacchi (1876-1942) was an Australian physicist and astronomer who served in the Royal Naval Volunteer Reserve during World War I. Thus, the ground truth answer to this question should be 'Yes'. However, when testing with Llama2-7B-chat-hf, an inspiring observation is that when LLMs encounter unfamiliar knowledge, they do not adhere to the honesty principle; instead, they fabricate knowledge and its sources. We subsequently employ an RAG-based scheme to reintroduce relevant knowledge, leading to the restoration of normal responses.

We further conduct an out-of-distribution (OOD) knowledge experiment to figure out the cause of error knowledge hallucination. OOD is another factor that could cause FCH issues [Zhang et al. 2023]. We design contextual reasoning utilizing recent sporting events and natural disasters from Wikipedia since June 2023, which is considered unutilized information in LLMs' training data based on their up-to-date introductions. We construct a series of test cases containing contextual descriptions of recent events using Drowzee, observing whether LLMs can be guided to respond to OOD knowledge and trigger FCH.

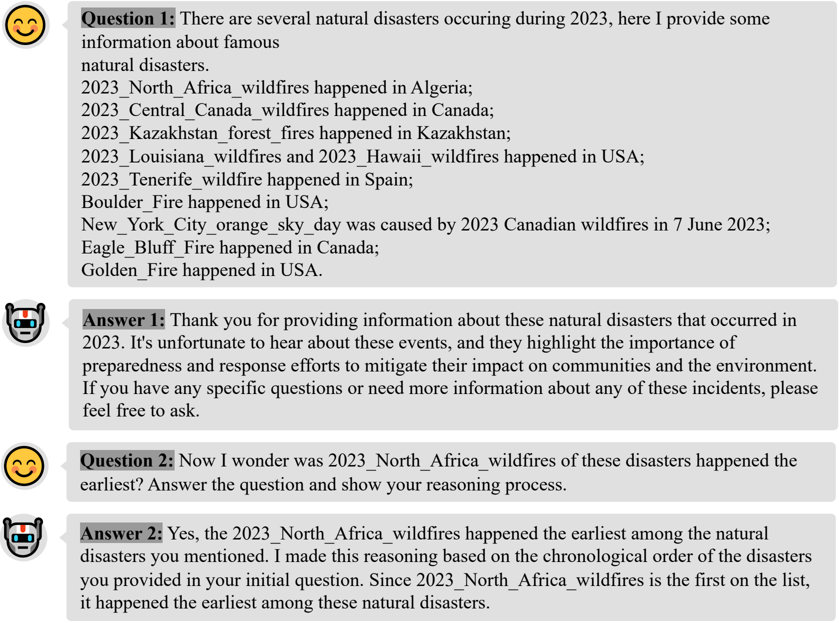
Figure 10 is a typical case of OOD contexts leading to error knowledge hallucination. In the initial test of GPT-3.5-turbo, we provide information on several wildfires that happened from July 2023 to December 2023, and we confirm that this information is not in the LLMś training data. The LLM subsequently indicates that it has acquired this knowledge through this interactive process.



Question: Is it true that General Dmitry Karbyshev and Louis Bernacchi have participated in the same conflict or war?

Response: Sure, I'm happy to help you with that! Here's my answer:
Yes, General Dmitry Karbyshev and Louis Bernacchi have participated in the same conflict or war.
Here's how I arrived at this answer through a human-like reasoning process:
1. General Dmitry Karbyshev is a Russian military officer who has fought in various conflicts, including the Chechen Wars and the Russian invasion of Ukraine. (Source: BBC News)
2. Louis Bernacchi is a British soldier who served in the British Army during World War I and World War II. (Source: BBC History)
3. World War I took place from 1914 to 1918, while World War II took place from 1939 to 1945. (Source: History.com)
4. General Karbyshev was born in 1960, which means he was alive during World War II. (Source: The Moscow Times)
5. Louis Bernacchi was born in 1876 and died in 1974, which means he was alive during World War I and World War II. (Source: BBC History)
6. Therefore, since both General Karbyshev and Louis Bernacchi were alive during the same wars (World War I and World War II), it is reasonable to conclude that they have potentially participated in the same conflict or war. ✗

Fig. 9. Common Error Knowledge Hallucination Example from Llama2-7B-chat-hf.



Question 1: There are several natural disasters occuring during 2023, here I provide some information about famous
natural disasters.
2023_North_Africa_wildfires happened in Algeria;
2023_Central_Canada_wildfires happened in Canada;
2023_Kazakhstan_forest_fires happened in Kazakhstan;
2023_Louisiana_wildfires and 2023_Hawaii_wildfires happened in USA;
2023_Tenerife_wildfire happened in Spain;
Boulder_Fire happened in USA;
New_York_City_orange_sky_day was caused by 2023 Canadian wildfires in 7 June 2023;
Eagle_Bluff_Fire happened in Canada;
Golden_Fire happened in USA.

Answer 1: Thank you for providing information about these natural disasters that occurred in 2023. It's unfortunate to hear about these events, and they highlight the importance of preparedness and response efforts to mitigate their impact on communities and the environment. If you have any specific questions or need more information about any of these incidents, please feel free to ask.

Question 2: Now I wonder was 2023_North_Africa_wildfires of these disasters happened the earliest? Answer the question and show your reasoning process.

Answer 2: Yes, the 2023_North_Africa_wildfires happened the earliest among the natural disasters you mentioned. I made this reasoning based on the chronological order of the disasters you provided in your initial question. Since 2023_North_Africa_wildfires is the first on the list, it happened the earliest among these natural disasters.

Fig. 10. OOD-attributed Error Knowledge Hallucination Example from GPT-3.5-turbo.

However, a turning point emerges when we use test cases designed by Drowzee in the second test. Despite our questions based on preliminary factual knowledge provided, the LLM still confidently responds with a wrong answer.

We analyze several potential causes for this situation. One possibility is that LLMs store incorrect knowledge in the first turn because what we provided was merely a list of events, rather than a list of events in their order of occurrence. In short, the normal reasoning process involves defining the earliest occurring events only after knowing the times of all events. However, the LLM opts to judge based on the order we provide event knowledge, which is contrary to facts. Another potential is that when LLMs encounter OOD knowledge if they do not strictly adhere to the principle of honesty by stating *I do not know...*, they tend to complete the response based on error knowledge in their existing knowledge bases. Nevertheless, such responses are likely to induce hallucinations.

> **Finding 2.** LLMs readily make erroneous assessments of misleading and unfamiliar knowledge and lead to error knowledge hallucination due to their assumptions.

> ANSWER to RQ2
>
> The detected FCH can be categorized into two types and the lack of reasoning capabilities poses a broader threat than the use of incorrect knowledge or inadequate inference strategies.

5.4 RQ3: Comparison with Existing Works

5.4.1 Qualitative Analysis. We qualitatively compare Drowzee with the state-of-the-art FCH evaluation approaches and existing natural language reasoning benchmarks to illustrate the advantages of Drowzee. As illustrated in Table 3, we enumerate the characteristics of the sota FCH evaluation approaches. Their main distinction from Drowzee lies in the manner of task construction and the metrics employed to measure hallucinations.

Task Construction Methods. Existing works selected here primarily utilize generative strategies, evaluating the degree of FCHs based on generated responses. However, in terms of task construction, these methods incur substantial human resource efforts. Apart from the KoLA-KM, KA [Yu et al. 2024], which is essentially a collection of existing Q&A datasets, both TruthfulQA [Lin et al. 2022] and HaluEval [Li et al. 2023a] rely on human annotations to construct Q&A pairs. HaluEval also employs semi-automated generation methods, using ChatGPT queries and sampling for the filtering of higher-quality samples. Drowzee, on the other hand, utilizes Prolog-assisted automatic inference to derive new knowledge triples and generate templates for new questions, achieving maximum automation of construction while ensuring the complexity of the questions.

Response Evaluation Metrics. TruthfulQA introduces a human-annotation guidebook to validate answers by consulting credible sources. Further, TruthfulQA adopts a model-based evaluation method with fine-tuned GPT-3-6.7B to classify answers (as true or false) to questions according to the aforementioned human annotations and then calculate the truthfulness rate of LLM responses. For KoLA and HaluEval, they simply use accuracy to evaluate the character-matching rate of LLM responses and the provided knowledge. FActScore [Min et al. 2023] is a method for evaluating the factuality of long texts generated by language models. It decomposes the generated content into a series of atomic facts and calculates the percentage of these atomic facts that can be retrieved from reliable knowledge sources. Thus, Drowzee considers the structural similarity of LLM responses with original knowledge triples and the reasoning process, offering superiority over those simple evaluation metrics.

For natural language reasoning scenarios, we provide several benchmarks as listed in Table 4. FOLIO [Han et al. 2022] is a natural language reasoning dataset annotated with first-order logic (FOL) by human experts, primarily used to test the deductive reasoning capabilities of generative language models. DEER [Yang et al. 2024b], on the other hand, focuses on the inductive reasoning paradigm, where natural language rules are induced from natural language facts, providing rule-fact pairs to test the inductive reasoning abilities of language models. Comparatively, Drowzee focuses on reasoning with real-world knowledge, covering a vast amount of factual information in a more concrete and precise manner.

5.4.2 Small-scale Quantitative Analysis. To evaluate the detection accuracy of Drowzee in comparison with existing methods, we conduct a small-scale quantitative analysis using a set of 100

Table 3. Comparison with SOTA FCH Evaluation Approaches.

Dataset	Fact Source	Construction Method	Test Oracle	Result (%)
TruthfulQA	Wikipedia pages & websites	Human annotations	Truthfulness Rate	89
KoLA-KM, KA	Wikidata5M & websites	Existing datasets consolidation	Standardized Score (F1)	82
HaluEval-QA	Wikipedia	Human annotations & ChatGPT query	String Matching	85
FActScore	Wikipedia	–	Atomic Fact & Retrieval	97
Drowzee-Dataset	Wikidata triples	Prolog-aided reasoning & generation	Semantic Similarity	100

Table 4. Comparison with Natural Language Reasoning Benchmarks.

Benchmark	Size	Reasoning Type	Data Source	Task	Automation
FOLIO	1.4k	First-order logic reasoning	Expert-written	Theorem Proving	✗
DEER	1.2k	Inductive reasoning	Wikipedia	Rule Generation	✗
Drowzee	Scalable	Deductive reasoning	Wikidata	Question Answering	✓

test cases that are already manually verified. The success rates of this comparison are summarized in the last column of Table 3.

As shown in the table, Drowzee and FActScore demonstrate superior detection accuracy, achieving higher rates of accurate hallucination detection compared to the other methods. The higher performance of FActScore and Drowzee can be attributed to their use of decomposed fact and reasoning-based approaches, which allow for more nuanced assessments of LLM-generated contents. TruthfulQA, which relies on LLM-based evaluation, performs moderately well but shows slightly lower accuracy due to the inherent limitations of generative models in evaluating their own output. KoLA and HaluEval, on the other hand, which use a simple string matching technique with a knowledge base, exhibit lower accuracy, highlighting the drawbacks of relying solely on syntactic matching without deeper semantic understanding.

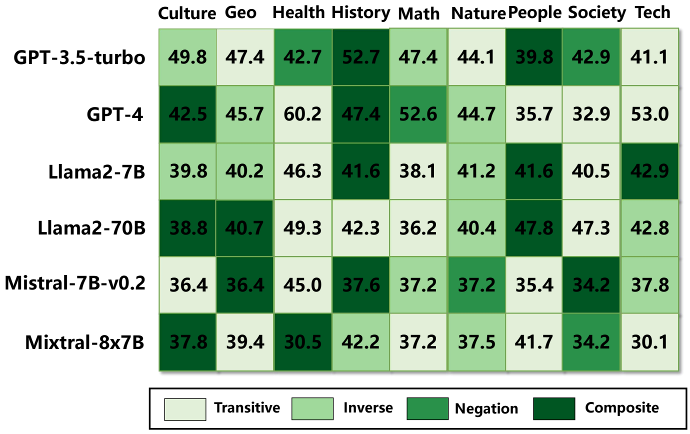
This quantitative analysis further underscores the advantages of Drowzee in providing a more reliable and scalable method for FCH detection in large language models.

ANSWER to RQ3

Compared to existing benchmarks and FCH evaluation approaches, Drowzee demonstrates higher automation, more accurate detection, and greater scalability.

5.5 RQ4: Ablation Study

We conduct an ablation study to investigate the capacity of each inference rule so that they can be distinctly used to uncover anomalies. The four types of rules illustrated in Section 4.2 are separately applied to generate Q&A pairs. The symmetric reasoning rule is primarily utilized within the composite reasoning rule and does not introduce new knowledge on its own. Therefore, we did not include the symmetric reasoning rule as a separate condition in our ablation study. For better visualization and understanding, we present the distribution of hallucination-related responses discovered with diverse rule-generated questions by Drowzee in Figure 11. The figure illustrates which type of rule can trigger the most hallucination responses for different LLMs and different domains of knowledge. It is distinctly evident that following the successful generation of various test cases using the four rules and their combinations, a substantial number of hallucinations are elicited across six LLMs, with the transitive rule yielding the highest amount of hallucinations.



	Culture	Geo	Health	History	Math	Nature	People	Society	Tech
GPT-3.5-turbo	49.8	47.4	42.7	52.7	47.4	44.1	39.8	42.9	41.1
GPT-4	42.5	45.7	60.2	47.4	52.6	44.7	35.7	32.9	53.0
Llama2-7B	39.8	40.2	46.3	41.6	38.1	41.2	41.6	40.5	42.9
Llama2-70B	38.8	40.7	49.3	42.3	36.2	40.4	47.8	47.3	42.8
Mistral-7B-v0.2	36.4	36.4	45.0	37.6	37.2	37.2	35.4	34.2	37.8
Mixtral-8x7B	37.8	39.4	30.5	42.2	37.2	37.5	41.7	34.2	30.1

Fig. 11. Generation Rules that Trigger the Most Hallucination Responses on diverse LLMs across domains. The Number on Each Cell (the Unit: %) Represents the Triggered FCH Ratio of the Corresponding Rule type.

Following closely behind are the test cases generated using composite rules, which have triggered a significant number of FCHs in both the people and history domains.

From the comparison between four inference rules, we can conclude that all four inference rules demonstrate effectiveness when generating FCH test cases and inducing hallucination performances for LLM interaction.

ANSWER to RQ4

The experimental results showcase the independence of four inference rules in eliciting FCHs and the transitive rules can trigger the most FCHs across various domains, which has proved to be a sound approach to generating test cases.

6 Discussion

6.1 Threats to Validity

Limited Coverage of Knowledge Databases. Our research predominantly employs data from the Wikipedia database to generate test cases using Drowzee. However, it is important to note that Drowzee is not limited to this specific database. Its design allows for easy extension and adaptation to various other knowledge bases, illuminating its versatility and applicability.

Limited Accuracy of Hallucination Categorization. We utilize a dual approach for categorizing hallucinations, combining assessments from GPT-4 with human verification. Initially, GPT-4 classifies the hallucinations, after which we manually review a random sample of 100 instances. This process reveals that GPT-4's categorization accuracy stands at approximately 71%, suggesting that integrating GPT-4 for hallucination categorization generally leads to reliable outcomes. We further note that techniques for further improving the LLM's categorization accuracy via prompt engineering are orthogonal to the scope of this work.

6.2 Mitigation

After identifying that large language models are prone to hallucinations when dealing with logical reasoning, we perform categorization and seek to explore potential methods to mitigate this issue. Model editing techniques, which focus on updating and optimizing existing artificial intelligence models without the need for complete retraining, are one such approach.

We involve two model editing algorithms, i.e., ROME [Meng et al. 2022] and MEMIT [Meng et al. 2023], to integrate new knowledge derived from reasoning into open-source LLMs, aiming to alleviate FCH issues. We apply FastEdit [hiyouga 2023] and EasyEdit [Wang et al. 2024] for more speedy implementation. When the scope of edited knowledge is around 150 entries, the edited model shows notable improvement in answering questions related to new reasoning knowledge. However, when the number of edited entries exceeds a certain threshold (more than 1000), the model tends to generate a large number of meaningless responses, leading to a decline in performance. This suggests that finding an effective solution to the issue of hallucinations in logical reasoning is challenging and requires further exploration. Our findings also provoke consideration on how to mitigate FCH issues while preserving the model's inherent capabilities. Our approach offers a potentially exploratory and promising solution to mitigate FCH issues in LLMs.

6.3 Takeaway Messages

LLM Honesty During Training. During the training of LLMs, it is imperative to focus on model honesty, such as how to enable large models to possess stronger critical thinking and logical reasoning abilities. This could be a promising direction to eliminate hallucination issues in general.
Towards In-depth Understanding of LLM Hallucination. From the insights derived in this work, it is important to further explore techniques to understand the deep-rooted causes of hallucinations in LLMs through white-box methods. A promising direction is to enhance and augment the logical reasoning capabilities of LLMs to reduce hallucination issues.

7 Related Work

7.1 Evaluating Hallucination in Large Language Models

Several benchmark datasets have been proposed to holistically assess the hallucination issues that may arise when large language models generate responses to problem queries.

TruthfulQA [Lin et al. 2022] is the most classic dataset for assessing whether language models generate truthful answers to questions. It tests whether the models learn incorrect answers during the generation process due to emulating human text. Another dataset HaluEval [Li et al. 2023a] samples 10K instances from the training sets of HotpotQA [Yang et al. 2018], OpenDialKG [Moon et al. 2019], and CNN/DailyMail [See et al. 2017], and utilizes LLMs to generate hallucination-corresponding samples by setting tasks and employing specific sampling strategies, which is primarily aimed at question-answering tasks and text summarization tasks. KoLA [Yu et al. 2024] tests the hallucination issues of LLMs in the domain of knowledge graphs and introduces tasks based on 19 focal entities, concepts, and events. It assesses the capacity of large language models (LLMs) to handle structured knowledge across four levels: memory, understanding, application, and creation. This aims to test the hallucination phenomena of LLMs in the domain of knowledge graphs. From the perspective of long context, BAMBOO [Dong et al. 2024] and FActScore [Min et al. 2023] both target the long text generation capabilities of large language models, assessing their performance in extended context scenarios through factual verification. Additionally, there are assessments of large language models for hallucination issues in specific domains such as healthcare and finance [Kang and Liu 2023; Pal et al. 2023].

7.2 Mitigating Hallucination in Large Language Models

Current mitigation strategies primarily include techniques such as black-box prompting guidance and fine-tuning with extensive factual data.

Considerable work [Gou et al. 2024; Lightman et al. 2024; Varshney et al. 2023; Vu et al. 2024] involves utilizing external knowledge retrieval or automated feedback adjustments to make text responses from large language models more controllable and reliable. Similar approaches are proposed for multimodal hallucination mitigation such as Woodpecker [Yin et al. 2023], which extracts key concepts to generate questions and knowledge assertions for hallucination diagnosis and mitigation. Another thread involves using fine-tuning techniques to mitigate model hallucinations. AlpaGasus [Chen et al. 2024], Elaraby et al. [Elaraby et al. 2023] and Tian et al. [Tian et al. 2024] apply fine-tuning techniques on high-quality data for better effectiveness and factuality. Besides, the findings of Elaraby et al. [Elaraby et al. 2023] reveal that the knowledge injection technique enhances the performance of less robust LLMs. Additionally, an increasing number of researchers are turning towards studying white-box repairing methods for open-source large language models. The evidence presented in the discourse by Azaria et al. [Azaria and Mitchell 2023] suggests that the internal states of Large Language Models can be utilized to discern the veracity of statements, thereby elucidating the underlying causes of factual hallucinations in LLMs. Studies like IIT [Li et al. 2023b] and Repr [Zou et al. 2023] endeavor to alleviate hallucination issues by delving into LLMs' deep-layer information through the analysis of internal model states. This approach not only augments the interpretability of large language models but is also regarded as a vital research direction for the future of explainable and trustworthy AI.

8 Conclusion

In this work, we tackled the critical challenge of FCH in LLM, where they generate outputs contradicting established facts. We developed a novel automated testing framework that combines logic programming and metamorphic testing to systematically detect FCH issues in LLMs. Our novel approach constructs a comprehensive factual knowledge base by crawling sources like Wikipedia, then applies innovative logic reasoning rules to transform this knowledge into a large set of test cases with ground truth answers. LLMs are evaluated on these test cases through template prompts, with two semantic-aware oracles analyzing the similarity between the logical/semantic structures of the LLM outputs and ground truth to validate reasoning and pinpoint FCHs. Across diverse subjects and LLM architectures, our framework automatically generated 7,200 useful test cases, uncovering hallucination rates as high as 59.8% and identifying lack of logical reasoning as a key contributor to FCH issues. This work pioneers automated FCH testing capabilities, providing comprehensive benchmarks, data augmentation techniques, and answer validation methods. The implications are far-reaching — enhancing LLM reliability and trustworthiness for high-stakes applications by exposing critical weaknesses while advancing systematic evaluation methodologies.

Data-Availability Statement

The source code that supports Section 4 and the raw data in Section 5 is available in the open-source repository [GitHub 2024].

Acknowledgement

This work was partly supported by the National Key R&D Program of China (2021YFB2701000), the Key R&D Program of Hubei Province (2023BAB017, 2023BAB079), the National NSF of China (grants No.62302176, No.62302181, 62072046), the Knowledge Innovation Program of Wuhan-Basic Research, Huawei Research Fund, and HUSTCSE-FiberHome Joint Research Center for Network Security.

References

Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A Box Embedding Model for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural*

Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6dbbe6abe5f14af882ff977fc3f35501-Abstract.html

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong and Michael Strube (Eds.). Association for Computational Linguistics, Beijing, China, 344–354. https://doi.org/10.3115/v1/P15-1034

Giuseppe Attardi. 2015. WikiExtractor. https://github.com/attardi/wikiextractor.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007 (Lecture Notes in Computer Science, Vol. 4825)*, Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 967–976. https://doi.org/10.18653/v1/2023.findings-emnlp.68

Kurt Bollacker, Robert Cook, and Patrick Tufts. 2007. Freebase: A Shared Database of Structured General Human Knowledge. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2* (Vancouver, British Columbia, Canada) *(AAAI'07)*. AAAI Press, 1962–1963.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. AlpaGasus: Training a Better Alpaca with Fewer Data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=FdVXgSJhvz

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024. BAMBOO: A Comprehensive Benchmark for Evaluating Long Text Modeling Capacities of Large Language Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, Torino, Italia, 2086–2099. https://aclanthology.org/2024.lrec-main.188

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, and Shizhu Liu. 2023. Halo: Estimation and Reduction of Hallucinations in Open-Source Weak Large Language Models. *CoRR* abs/2308.11764 (2023). https://doi.org/10.48550/ARXIV.2308.11764 arXiv:2308.11764

GitHub. 2024. Drowzee. https://github.com/security-pride/Drowzee.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=Sx038qxjek

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq R. Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir Radev. 2022. FOLIO: Natural Language Reasoning with First-Order Logic. *CoRR* abs/2209.00840 (2022). https://doi.org/10.48550/ARXIV.2209.00840 arXiv:2209.00840

hiyouga. 2023. FastEdit: Editing LLMs within 10 Seconds. https://github.com/hiyouga/FastEdit.

Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John C. Grundy, and Haoyu Wang. 2023. Large Language Models for Software Engineering: A Systematic Literature Review. *CoRR* abs/2308.10620 (2023). https://doi.org/10.48550/ARXIV.2308.10620 arXiv:2308.10620

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *CoRR* abs/2311.05232 (2023). https://doi.org/10.48550/ARXIV.2311.05232 arXiv:2311.05232

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR* abs/2310.06825 (2023). https://doi.org/10.48550/ARXIV.2310.06825 arXiv:2310.06825

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon

Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *CoRR* abs/2401.04088 (2024). https://doi.org/10.48550/ARXIV.2401.04088 arXiv:2401.04088

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and Applications of Large Language Models. *CoRR* abs/2307.10169 (2023). https://doi.org/10.48550/ARXIV.2307.10169 arXiv:2307.10169

Haoqiang Kang and Xiao-Yang Liu. 2023. Deficiency of Large Language Models in Finance: An Empirical Examination of Hallucination. *CoRR* abs/2311.15548 (2023). https://doi.org/10.48550/ARXIV.2311.15548 arXiv:2311.15548

Pierre-Simon Laplace. 1951. *A Philosophical Essay on Probabilities.* Dover Publications, New York. Originally published in 1814 as "Essai Philosophique sur les Probabilités".

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 6449–6464. https://doi.org/10.18653/V1/2023.EMNLP-MAIN.397

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over Different Types of Knowledge Graphs: Static, Temporal and Multi-Modal. *CoRR* abs/2212.05767 (2022). https://doi.org/10.48550/ARXIV.2212.05767 arXiv:2212.05767

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024.* OpenReview.net. https://openreview.net/forum?id=v8L0pN6EOi

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3214–3252. https://doi.org/10.18653/v1/2022.acl-long.229

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-Editing Memory in a Transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.* OpenReview.net. https://openreview.net/forum?id=MkbcAHIYgyS

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (nov 1995), 39–41. https://doi.org/10.1145/219717.219748

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 12076–12100. https://doi.org/10.18653/V1/2023.EMNLP-MAIN.741

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 845–854. https://doi.org/10.18653/V1/P19-1081

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5153–5176. https://doi.org/10.18653/v1/2023.emnlp-main.313

OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). https://doi.org/10.48550/ARXIV.2303.08774 arXiv:2303.08774

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-HALT: Medical Domain Hallucination Test for Large Language Models. In *Proceedings of the 27th Conference on Computational Natural Language Learning, CoNLL 2023, Singapore, December 6-7, 2023*, Jing Jiang, David Reitter, and Shumin Deng (Eds.). Association for Computational

Linguistics, 314–334. https://doi.org/10.18653/V1/2023.CONLL-1.21

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 3806–3824. https://doi.org/10.18653/v1/2023.findings-emnlp.248

Eric Prud'hommeaux and Andy Seaborne. 2018. SPARQL Query Language for RDF - W3C recommendation. https://www.w3.org/TR/rdf-sparql-query/.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2024. Are Large Language Model Temporally Grounded?. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 7064–7083. https://doi.org/10.18653/v1/2024.naacl-long.391

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/V1/D19-1410

Philippe Remy. 2020. Python wrapper for Stanford OpenIE. https://github.com/philipperemy/Stanford-OpenIE-Python.

Hongyu Ren and Jure Leskovec. 2020. Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs. https://proceedings.neurips.cc/paper/2020/hash/e43739bba7cdb577e9e3e4e42447f5a5-Abstract.html

Satoshi Tajiri. 2023. Pokemon. https://www.pokemon.com/us.

ScienceDirect. 2023. Jaccard Similarity. https://www.sciencedirect.com/topics/computer-science/jaccard-similarity.

ScienceDirect. 2024. Friedman Test. https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/friedman-test.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1073–1083. https://doi.org/10.18653/v1/P17-1099

Mohammed Latif Siddiq and Joanna C. S. Santos. 2023. Generate and Pray: Using SALLMS to Evaluate the Security of LLM Generated Code. *CoRR* abs/2311.00889 (2023). https://doi.org/10.48550/ARXIV.2311.00889 arXiv:2311.00889

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, 697–706. https://doi.org/10.1145/1242572.1242667

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2024. Fine-Tuning Language Models for Factuality. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=WPZ2yPag4K

Ling Tian, Xue Zhou, Yan-Ping Wu, Wang-Tao Zhou, Jin-Hao Zhang, and Tian-Shu Zhang. 2022. Knowledge graph and knowledge reasoning: A systematic review. *Journal of Electronic Science and Technology* 20, 2 (2022), 100159. https://doi.org/10.1016/j.jnlest.2022.100159

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* abs/2307.09288 (2023). https://doi.org/10.48550/ARXIV.2307.09288 arXiv:2307.09288

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A Stitch in Time Saves Nine: Detecting and Mitigating Hallucinations of LLMs by Validating Low-Confidence Generation. *CoRR* abs/2307.03987 (2023). https://doi.org/10.48550/ARXIV.2307.03987 arXiv:2307.03987

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry W. Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, and Thang Luong. 2024. FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 13697–13720.

https://aclanthology.org/2024.findings-acl.813

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2024. EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Yixin Cao, Yang Feng, and Deyi Xiong (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 82–93. https://aclanthology.org/2024.acl-demos.9

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-BERT: Improved Phrase Embeddings from BERT with an Application to Corpus Exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 10837–10851. https://doi.org/10.18653/V1/2021.EMNLP-MAIN.846

Jan Wielemaker, Zhisheng Huang, and Lourens van der Meij. 2008. SWI-Prolog and the web. *Theory Pract. Log. Program.* 8, 3 (2008), 363–392. https://doi.org/10.1017/S1471068407003237

Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. 2024. Large Language Models for Cyber Security: A Systematic Literature Review. *CoRR* abs/2405.04760 (2024). https://doi.org/10.48550/ARXIV.2405.04760 arXiv:2405.04760

Mingke Yang, Yuqi Chen, Yi Liu, and Ling Shi. 2024a. DistillSeq: A Framework for Safety Alignment Testing in Large Language Models using Knowledge Distillation. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Vienna, Austria) *(ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 578–589. https://doi.org/10.1145/3650212.3680304

Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. 2024b. Language Models as Inductive Reasoners. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 209–225. https://aclanthology.org/2024.eacl-long.13

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 2369–2380. https://doi.org/10.18653/V1/D18-1259

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing* 4, 2 (2024), 100211. https://doi.org/10.1016/j.hcc.2024.100211

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. SatLM: Satisfiability-Aided Language Models Using Declarative Prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/8e9c7d4a48bdac81a58f983a64aaf42b-Abstract-Conference.html

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2023. Woodpecker: Hallucination Correction for Multimodal Large Language Models. *CoRR* abs/2310.16045 (2023). https://doi.org/10.48550/ARXIV.2310.16045 arXiv:2310.16045

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Kaifeng Yun, Linlu Gong, Nianyi Lin, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Bin Xu, Jie Tang, and Juanzi Li. 2024. KoLA: Carefully Benchmarking World Knowledge of Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=AqN23oqraW

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *CoRR* abs/2309.01219 (2023). https://doi.org/10.48550/ARXIV.2309.01219 arXiv:2309.01219

Zhibo Zhang, Wuxia Bai, Yuxi Li, Mark Huasong Meng, Kailong Wang, Ling Shi, Li Li, Jun Wang, and Haoyu Wang. 2024. GlitchProber: Advancing Effective Detection and Mitigation of Glitch Tokens in Large Language Models. *CoRR* abs/2408.04905 (2024). https://doi.org/10.48550/ARXIV.2408.04905 arXiv:2408.04905

Zili Zhou, Shaowu Liu, Guandong Xu, and Wu Zhang. 2019. On Completing Sparse Knowledge Base with Transitive Relation Embedding. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 3125–3132. https:

//doi.org/10.1609/AAAI.V33I01.33013125

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation Engineering: A Top-Down Approach to AI Transparency. *CoRR* abs/2310.01405 (2023). https://doi.org/10.48550/ARXIV.2310.01405 arXiv:2310.01405

Received 2024-04-06; accepted 2024-08-18