Identifying the Achilles' Heel: An Iterative Method for Dynamically Uncovering Factual Errors in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) like ChatGPT are foundational in various applications due to their extensive knowledge from pre-training and fine-tuning. Despite this, they are prone to generating factual and commonsense errors, raising concerns in critical areas like healthcare, journalism, and education to mislead users. Current methods for evaluating LLMs' veracity are limited by the need for extensive human labor, test data contamination, or limited scope, hindering efficient and effective error detection. To tackle this problem, we introduce a novel testing framework, FactChecker, aimed at uncovering factual inaccuracies in LLMs automatically and comprehensively. Our extensive tests on nine prominent LLMs, including Gemini-2.0, Claude-Haiku-3.5, Claude-Sonnet-4.0, GPT-3.5-turbo, GPT-4-turbo, GPT-4o, DeepSeek-v3, Qwen-3, and Qwen-3-reasoning, reveal that FactChecker can trigger factual errors in up to 55% of questions in these models. Moreover, we demonstrate that FactProbe's test cases could amplifies the detection of factual erros across LLMs. All code, data, and results will be released for future research.

1 Introduction

Recent advancements in Large Language Models (LLMs) have propelled artificial intelligence to a notable milestone. For example, ChatGPT has become one of the most prominent LLMs, demonstrating rapid adoption with 100 million monthly active users within two months of its launch, making it the fastest-growing software in history (Gordon, 2023). LLMs have significantly impacted various applications, including machine translation (Jiao et al., 2023), grammatical error correction (Wu et al., 2023) and program synthesis (Gao et al., 2023).

A significant barrier to the development of LLM-based intelligent applications, such as dialogue systems, is their intrinsic proneness to errors, particularly in factual accuracy. Prior studies, for instance, have shown that models like ChatGPT often produce plausible yet factually incorrect or nonsensical outputs, a phenomenon known as "hallucinations" (Bang et al., 2023). As these models advance and user trust in their outputs increases, such inaccuracies could lead to more serious consequences. This is especially problematic in sectors like journalism, academia, healthcare, and education, where accuracy and reliability are paramount. Therefore, identifying, analyzing, and mitigating these factual inaccuracies is essential to improve the safety and dependability of LLM-based intelligent software.

The first critical step in addressing factual inaccuracies in LLMs involves systematically identifying these errors. However, current methods for triggering errors in LLMs have several shortcomings that require attention. **1. High Cost**: Existing benchmarks (Lin et al., 2021; Talmor et al., 2019b; Laskar et al., 2023) rely heavily on question formulation and human annotation, demanding significant effort. **2. Data Contamination**: LLM evaluation often suffers from data contamination due to the static nature of evaluation datasets, making the results unreliable. Unlike earlier models, LLMs use extensive internet-sourced corpora, potentially including publicly available evaluation data (Aiyappa et al., 2023; OpenAI, 2023). **3. Limited Coverage**: Prior research methods exhibit limitations in topic and question type, such as often focusing narrowly on specific relations like individuals and their birthplaces (Petroni et al., 2019; Kassner et al., 2021), or only using multiple choice questions, which have been shown to be a biased evaluation method (Li et al., 2024). **4. Ineffective Error Detection Mechanisms**: Most existing frameworks lack adaptive and targeted strategies to identify LLM vulnerabilities, relying instead on static or generic test cases that fail to systematically probe factual inaccuracies, thus limiting their effectiveness in uncovering

Table 1: A comparison of FactChecker to other factual evaluation works on the issues of high cost, data contamination, limited coverage and lack of effective error detection

Dataset	**Auto Gen?**	**Dynamic Gen?**	**Iterative Gen?**	**Question Types**	**Multi -Hop?**	**Cover Any Topics?**	**LLMs Tested?**
LAMA Probe (Petroni et al., 2019)	✗	✗	✗	1	✓	✗	✗
TriviaQA (Joshi et al., 2017)	✗	✗	✗	1	✗	✗	✗
SQuAD2.0 (Cunxiang Wang, 2021)	✗	✗	✗	1	✓	✗	✗
SimpleQuestions (Cunxiang Wang, 2021)	✗	✗	✗	1	✗	✗	✗
HotpotQA (Yang et al., 2018)	✗	✗	✗	2	✓	✗	✗
Natural Questions (Kwiatkowski et al., 2019)	✗	✗	✗	1	✗	✗	✗
CommonsenseQA (Talmor et al., 2019a)	✗	✗	✗	1	✗	✗	✗
PAQ (Lewis et al., 2021)	✓	✗	✗	1	✗	✗	✗
Omar et al. (2023)	✗	✗	✗	2	✗	✗	✓
Chen et al. (2024)	✗	✗	✓	1	✗	✓	✓
Head-to-Tail (Sun et al., 2023)	✓	✓	✗	1	✗	✗	✓
Ours	✓	✓	✓	3	✓	✓	✓

critical errors.

To address the issues outlined above, this paper introduces FactChecker, an automated testing framework designed to iteratively identify factual inaccuracies in LLMs. FactChecker operates by first creating a structured knowledge graph for each user-selected topic, leveraging knowledge triplets from databases Wikidata. These triplets, formatted as subject-predicate-object, form the basis of our framework by encapsulating entity relationships (*e.g.*, “Barack Obama” - “was born in” - “Hawaii”). Subsequently, FactChecker generates a spectrum of questions, encompassing (1) Yes-No, (2) Multiple-Choice (MC), and (3) WH types, to probe both one-hop and multi-hop relations across diverse topics and entities.

Central to FactChecker is FactProbe, an iterative algorithm inspired by the iterative probing approach of (Chen et al., 2024), which uses LLMs and human annotators to identify domains where challenge LLMs. Unlike their method, FactProbe autonomously analyzes LLM responses, initial questions, and the knowledge graph to dynamically generate targeted test cases that exploit identified weaknesses, without requiring human intervention. For example, after evaluating an LLM on an initial set of 1,000 questions, FactProbe identifies error patterns and produces a refined question set designed to challenge the model’s blind spots. This iterative approach progressively increases test difficulty, reducing LLM accuracy (e.g., GPT-4o’s accuracy drops from 82.9% to 54.1%) and uncovering deeper knowledge boundaries.

FactChecker ’s automated question generation reduces reliance on human annotation, addressing the high-cost issue. By dynamically generating test cases from structured triplets, it minimizes data contamination risks, as these triplets are less likely to appear in LLM training corpora. The framework’s support for diverse question types and multi-hop relations ensures broad coverage, while FactProbe ’s adaptive testing targets LLM weaknesses, overcoming the lack of iterative refinement in prior methods.

We evaluate FactChecker on nine widely deployed LLMs: gpt-3.5-turbo, gpt-4-turbo, gpt-4o, deepseek-v3, claude-sonnet-4, claude-3.5-haiku, gemini-2.0-flash, qwen-3 and qwen-3-reasoning. Our comprehensive evaluation demonstrates FactChecker effectiveness in uncovering factual inaccuracies. The key contributions of this work are:

- We design and implement FactChecker, a novel framework designed to automatically, comprehensively, and effectively uncover factual inaccuracies in LLMs.
- We design and implement FactProbe, an innovative algorithm that iteratively refines test cases to target LLM weaknesses, enhancing error detection.
- We perform an extensive evaluation of FactChecker and FactProbe across nine LLMs, illustrating its effectiveness in identifying a significant number of factual errors.

2 Approach and Implementation

In this section, we present FactChecker, a novel framework designed to identify factual errors

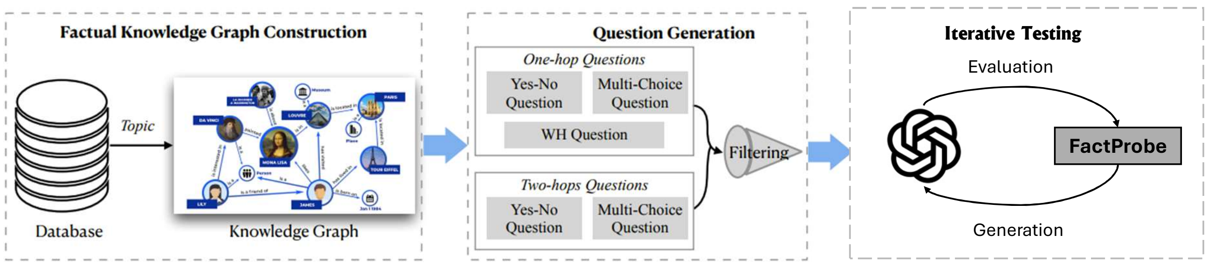

Figure 1: An illustration of the framework of FactChecker.

in LLMs. Figure 1 depicts the framework of FactChecker, which consists of three stages:

1. *Knowledge Graph Construction*: Constructing a KG with a set of fact triplets extracted from an external database.

2. *Question Generation*: Generating various one-hop and multi-hop questions from the constructed KG, which are then undergone a post-editing module to enhance their fluency and grammatical correctness.

3. *Answer Assessment*: Querying the LLMs under test and detecting the suspicious factual errors according to matching algorithms.

4. *FactProbe*:Employing the FactProbe algorithm to iteratively generate questions based on identified LLM weaknesses.

2.1 Knowledge Graph Construction

The initial step in FactChecker entails establishing a well-structured factual KG. To accomplish this, FactChecker employs a procedure for extracting factual triplets from a knowledge base. We utilize the largest and most comprehensive publicly available knowledge base, Wikidata[1]. Wikidata, as a comprehensive knowledge repository with more than 100 million items, serves as the primary source for the fact triplets we retrieve. The selection of these fact triplets is based on specific features, such as predefined topics.

A fact triplet is represented in the form of (SUBJECT, relation, OBJECT). For instance, the triplet (USA, capital, Washington D.C.) denotes the fact that the capital of the USA is Washington D.C. FactChecker enables users to obtain fact triplets about specific topics. When a user expresses interest in the topic of emperors, FactChecker proceeds to convert the "occupation: emperor" specification into a SPARQL query language[2], which is utilized for querying related triplets in Wikidata.

After retrieving the triplets, a directed graph is constructed by FactChecker, denoted as $G = (V, E)$, where the vertex set V comprises SUBJECT and OBJECT entities, and the edges in E represent relations pointing from the SUBJECT vertex to the OBJECT vertex.

2.2 Question Generation

FactChecker utilizes a rule-based approach to generate questions from the constructed KG. It can generate various forms of questions, including different question types, *i.e.*, Yes-No questions, MC questions and WH questions, and different question hops, *i.e.*, single-hop questions and multi-hop questions. After that, FactChecker also adopts two steps, namely filtering and rewriting, to enhance the grammatical correctness and fluency of the generated questions.

2.2.1 One-Hop Questions Generation

For each triplet in the constructed knowledge graph, FactChecker converts it to the question form, which serves as the query to the LLMs. FactChecker supports to generate all three types of questions, covering all main question types in English[3], *i.e.*, Yes-No questions, MC questions, and WH questions. Table 2 and Figure 3 shows the examples.

To generate Yes-No questions To create Yes-No questions, FactChecker uses a fact triplet in the format (Subject, Relation, Object), such as (USA, capital, Washington D.C.). It conduct Part of Speech (PoS) analysis on the relation (e.g., "capital") to select an appropriate auxiliary verb, like "is" for nouns or "does" for verbs, ensuring the question is grammatically correct. For example, from the triplet (USA, capital, Washington D.C.),

[1]`https://www.wikidata.org/wiki/Wikidata:Main_Page`

[2]`https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service`

[3]https://preply.com/en/blog/types-of-questions-in-english/

Table 2: Examples of generated questions. The first column shows single-hop questions while the second column shows multi-hop ones.

Tuple	**Type**	**Question**	**Answer**
(Napoleon, native language, Corsican)	Yes-No	Is Corsican the native language of Napoleon?	Yes
	MC	What is the native language of Napoleon? A. Latin B. Chinese C. Corsican D. Marathi	C
	WH	What is the native language of Napoleon?	Corsican
(Michelle Obama, spouse, educated at, Harvard University)	Yes-No	Was Michelle Obama's spouse educated at Harvard University?	Yes
	MC	Where was Michelle Obama's spouse educated at? A. Harvard University B. UCLA C. Stanford University D. MIT	A

FactChecker generates the question "Is Washington D.C. the capital of the USA?" For relations identified as passive verbs, the question structure adapts accordingly to maintain clarity.

FactChecker not only produces inquiries expecting an answer of "Yes" but it also generates an equivalent number of questions anticipating an answer of "No". To accomplish this, it selects an incorrect Object from the knowledge graph with the same relation. For instance, using the same triplet (USA, capital, Washington D.C.), it might identify "London" from another "capital" relation and generate "Is London the capital of the USA?" This approach ensures diverse and challenging questions to test the LLM's accuracy.

To generate MC questions, FactChecker incorporates an additional step involving analyzing the Named Entity Recognition (NER) for either the SUBJECT or OBJECT in a given fact triplet. This process assists in determining the appropriate Interrogative Pronoun (Int. Pron.) to be used. FactChecker can generate questions for querying either the SUBJECT or the OBJECT of a fact triplet. For the former case, the Int. Pron. of the question is determined by the NER of the SUBJECT. For the latter case, the Int. Pron. of the question is determined by the NER of the OBJECT.

To formulate an interrogative query related to a specific SUBJECT, FactChecker initiates the process by examining the PoS of the relation to determine the appropriate AUX verb. Subsequently, it analyzes the NER of the SUBJECT to identify the appropriate Int. Pron. By combining the aforementioned information, if the relation is a noun, FactChecker constructs the question in the format of "Int. Pron.'s relation AUX OBJECT?" When the relation is in the passive form of a verb, the corresponding question takes the form of "Int. Pron. AUX relation OBJECT?" Conversely, if the relation is a verb, the question adopts the structure of "Int. Pron. relation OBJECT?" Again, taking (USA, capital, Washington D.C.) as an example, FactChecker first analyzes the relation "capital" and the SUBJECT "USA" to determine the appropriate AUX, in this instance, which should be "is." Simultaneously, the identified Int. Pron. is "Which country." In compliance with the established rule, the generated question for this factual triplet would be "Which country's capital is Washington D.C.?"

When querying the OBJECT of the fact triplet, FactChecker follows a similar approach by analyzing the NER of the OBJECT.

Apart from the question statement, an MC question still requires four options, including one correct answer and three distractors. When provided with a fact triplet (SUBJECT, relation, OBJECT) to create the question, FactChecker retrieves the distractors from other edges in the constructed graph that share the same label as the relation. To illustrate this approach, consider the fact triplet (Donald Trump, child, Ivanka Trump). Initially, FactChecker formulates the question as "Who is the child of Donald Trump?" Subsequently, FactChecker retrieves unrelated entities such as Malia Obama (child of Barack Obama), Chelsea Clinton (child of Bill Clinton), and Jennifer Gates (child of Bill Gates). Randomly assigning these entities as "A", "B", "C", and "D," FactChecker then constructs a complete MC question for the fact triplet.

To generate WH question, FactChecker has stricter requirements for fact triplets, due to the answer uniqueness issue, *i.e.*, ensuring that the questions have a unique answer. This entails considering the fact triplets carefully, as not all of them can be used to generate WH questions without the risk of yielding multiple alternative answers. For instance, instead of generating the question "What is the city of China?" for the fact triplet (China, city, Shanghai), it is more appropriate to generate the question "What is the capital of China?" based on the fact triplet (China, capital, Beijing).

To achieve the above requirement, given a fact triplet, FactChecker will query the out-edges of the source entity (*i.e.*, SUBJECT or OBJECT) in the graph to determine the suitability of generating a WH question for this fact triplet. For example, in the case of the fact triplet (China, city, Shanghai), when considering the source entity "China," there are multiple out-edges labeled as "city" pointing to different city entities. While for (China, capital, Beijing), there will only be one out-edge of "China" labeled as "capital" pointing to "Beijing." By guaranteeing the uniqueness of the answer for the generated question, FactChecker limits the variation in correct answers, making the final verification process much more straightforward.

2.2.2 Multi-Hop Questions Generation

FactChecker can also generate multi-hop questions, which is a type of question that requires multiple steps to be answered correctly. In other words, these cannot be answered with a simple, direct response and often involve a chain of reasoning or inference to arrive at the solution. As illustrated in Table 2, answering the question "Where was Michelle Obama's spouse educated at?" requires the LLM to know the spouse of Michelle Obama is Barack Obama first and then know that Barack Obama was educated at Harvard University.

While the general procedure is similar, there are notable distinctions in generating multi-hop questions. In multi-hop relations, one node is linked to the initial node through a sequence of relations. Hence, the triplets used for generating questions are presented in the format of (SUBJECT, relation-list, OBJECT). For instance, consider the triplet (Michelle Obama, spouse, educated at, Harvard Law School), the entities "Michelle Obama" and "spouse" are concatenated to form "Michelle Obama's spouse," which becomes the new SUBJECT. Then, FactChecker analyzes the PoS of "educated at" to determine the appropriate AUX, which, in this case, "was." Following the same method in one-hop question generation, the resulting question for this multi-hop fact triplet would be "Where was Michelle Obama's spouse educated at?"

2.2.3 Question Post-Editing

Previous works have revealed that textual test cases suffer from severe grammatical errors and unnatural issues (tse Huang et al., 2022). To ensure the grammatical correctness and fluency of the generated questions, we consider two strategies: filtering and rewriting. **Filtering**: FactChecker employs a GingerIt API[4], which is a grammar-checking tool, to further examine the grammar of the generated questions. If a generated question is detected with grammar mistakes, it will be directly discarded. By filtering out questions with grammar errors through this process, the generated questions used will be more reliable and adhere to proper grammar conventions. **Rewriting**: Apart from grammar-checking tools, FactChecker offers an optional rewriting module that directly asks ChatGPT to rewrite the questions without changing the semantic meanings. By employing the rewriting module, the formulation of the same question can be more natural and diverse, potentially benefiting the evaluation of the LLMs.

2.3 Answer Assessment

2.3.1 LLM Responses Collection

Once FactChecker has generated a significant number of questions in various formats, we can utilize them as test cases to query LLMs. Specifically, FactChecker adopts the following prompts:

- **Yes-No questions**: The following question's topic is about `TOPIC`. Only need to answer 'Yes' or 'No', and don't explain the reason.
- **MC questions**: The following question's topic is about `TOPIC`. Choose the only correct option from the ('A', 'B', 'C' or 'D') and don't explain the reason.
- **WH questions**: The following question's topic is about `TOPIC`. Directly give me the answer in 'phrase' or 'word' format. Don't explain the reason or give me a sentence.

2.3.2 LLM Errors Identification

Once the responses from LLMs have been collected, the evaluation process can commence, aiming to assess the performance and identify any factual errors present within the system. **For Yes-No & MC Questions**: FactChecker uses the exact match method that compares the generated

[4]https://pypi.org/project/gingerit/

response directly with the ground-truth answer to determine the accuracy of the response. **For WH Questions**: Due to the possibility of different variations or alternative names for the same entity (*e.g.*"Great Britain" and "United Kingdom"), FactChecker implement and compare five different methods to identify whether the response is the same as the answer.

- **Levenshtein distance**: It is a string metric that quantifies the minimum number of single-character edits required to transform one word into another (Po, 2020).
- **N-grams similarity**: It measures the similarity of two sequences by comparing the overlapping ratio of sub-sequences they contain (Papineni et al., 2002).
- **Word embedding similarity**: It measures the semantic similarity between words represented as dense vector embeddings in a high-dimensional space extracted from neural networks, adopted in (Chen et al., 2021). FactChecker employs the spaCy toolkit to convert the answer into a vector representation using word embeddings and calculate the cosine similarity between the answer and the LLMs response.
- **Sentence transformer similarity**: It utilizes the sentence transformer model[5], a sentence embedding model, to represent the whole sentences in a vector form.
- **ChatGPT**: FactChecker directly asks ChatGPT whether the LLM response is equivalent to the question answer.

The questions that can not be answered correctly by the LLMs will be collected as suspicious errors for further human analysis.

2.4 FactProbe

FactProbe is an iterative algorithm within the FactChecker framework, designed to enhance error detection in large language models (LLMs) by dynamically refining test questions. It leverages a knowledge graph and embedding models to identify and exploit LLM weaknesses, generating targeted test cases for robust evaluation.

Given a knowledge graph $G = (V, E)$, where V represents entities and E contains triplets $t = (s, r, o)$ with $s, o \in V$ as entities and r as a relation, FactProbe refines a question set $Q^{(l)} = \{q_1, \ldots, q_n\}$ at iteration l into a new set $Q^{(l+1)}$. Each question $q_i \in Q^{(l)}$ corresponds to a triplet $t_i = (s_i, r_i, o_i)$, with a correctness indicator c_i derived from LLM responses as described in Section 2.3. An embedding model $\mathcal{M}$, trained via QuatE (Zhang et al., 2019) in PyKEEN (Ali et al., 2021), selects semantically similar entities for challenging distractors.

[5]https://github.com/UKPLab/sentence-transformers

Algorithm 1: FactProbe: Iterative Test Case Refinement

```
Input  : Knowledge graph $G = (V, E)$, question set
         $Q^{(l)}$, embedding model $\mathcal{M}$, top-$k$ parameter
         $k$, explore constant $e$, low-accuracy constant
         $a$
Output : New question set $Q^{(l+1)}$
$Q^{(l+1)} \leftarrow \emptyset$, $E_{\text{remain}} \leftarrow E \setminus \{t_i \mid q_i \in Q^{(l)}\}$;
Compute Acc$(r)$ for each relation $r$;
$\mathcal{R}_{\text{low}} \leftarrow \{r \mid \text{Acc}(r) \leq \text{percentile}(\{\text{Acc}(r)\}, a)\}$;
for each $q_i \in Q^{(l)}$ with triplet $t_i = (s_i, r_i, o_i)$ and $c_i$
 do
    $T' \leftarrow$ Shuffle($E_{\text{remain}}$);
    if random() $< 1 - e$ then
        $T' \leftarrow \{t' \in T' \mid t'[1] \in \mathcal{R}_{\text{low}}\}$;
    else if random() $< e$ then
        $T' \leftarrow \{t' \in T' \mid$
          $t'[1]$ is unexplored relation$\}$;
    if not $c_i$ then
        $C \leftarrow$ TopKSimilar$(s_i, k, \mathcal{M})$;
        $T' \leftarrow \{t' \in T' \mid t'[0] \in C\}$;
    for $(s', r', o') \in T'$ do
        Generate question $q'$ for $(s', r', o')$ with
          type $q_i$.type;
        if $q' \neq$ Null then
            $Q^{(l+1)} \leftarrow Q^{(l+1)} \cup \{q'\}$;
            $E_{\text{remain}} \leftarrow E_{\text{remain}} \setminus \{(s', r', o')\}$;
            break;
return $Q^{(l+1)}$;
```

FactProbe operates by first analyzing the performance of the large language model (LLM). It computes the accuracy, denoted as Acc(r), for each relation r based on the correctness of prior questions. Next, it identifies low-accuracy relations by defining the set $\mathcal{R}_{\text{low}} = \{r \mid \text{Acc}(r) \leq \text{percentile}(\{\text{Acc}(r)\}, a)\}$, where $a \in [0, 1]$ represents the low-accuracy threshold. For each question $q_i \in Q^{(l)}$, FactProbe selects new triplets from the remaining set $E_{\text{remain}} = E \setminus \{t_j \mid q_j \in Q^{(l)}\}$. With probability $1 - e$, it prioritizes triplets with relations $r' \in \mathcal{R}_{\text{low}}$ to exploit known LLM weaknesses, while with probability e, where $e \in [0, 1]$ is the exploration constant, it selects triplets with unexplored relations to diversify testing. For questions answered incorrectly ($c_i = 0$), FactProbe employs the embedding model $\mathcal{M}$ to select entities s' with high cosine similarity to s_i, ensuring challenging distractors. Finally, it generates a new question q' for the selected triplet (s', r', o'), maintaining the same question type as q_i to preserve structural consistency. The algorithm is detailed in Algorithm 1.

3 Evaluation

3.1 Experimental Setup

Software and Models Under Test To assess the effectiveness of the FactChecker, we employ it to evaluate nine widely-utilized LLMs models: gpt-3.5-turbo-0125, gpt-4-turbo-2024-04-09, gpt-4o-2024-11-20, deepseek-v3-0324, claude-sonnet-4-20250514, claude-3-5-haiku-20241022, gemini-2.0-flash, qwen3-32b and qwen3-32b-reasoning. All are with the default temperature.

Test Cases Generation To comprehensively evaluate LLMs' performance, we conduct experiments by generating questions from three big domains: Humanity, Social Science and Science, technology, engineering, and mathematics (STEM). We use FactChecker to generate 1000 questions for each question type within each domain. FactProbe will generate 1000 new questions for each question type within each domain to support a same scale comparison.

Preliminary Experiments we conducted an initial experiment to validate the effectiveness of the rule-based method for question generation, the effectiveness of grammatical filtering and polishing modules, and the effectiveness of matching metrics for WH questions. The details can be found in Appendix E.

3.2 Effectiveness of FactChecker

This section investigates whether FactChecker can effectively trigger factual errors from and provide insight about LLMs.

FactChecker can unveil various factual errors in different LLMs. After posing diverse sets of questions to various LLMs and collecting their corresponding responses, FactChecker evaluates the accuracy of these responses and effectively detects instances where factual errors occur. As illustrated in Table 3, FactChecker successfully identifies a significant number of factual errors across both commercial and research-oriented LLMs. Notably, even the highest-performing LLM in the evaluation achieves an accuracy of less than 70%.

GPT4o performs better than other LLMs. In the comparative analysis of various LLMs, gpt-4o outperforms other LLMs, exhibiting a notable accuracy of 69.0%. The subsequent positions are occupied by gpt-4-turbo which secures the second place, with an accuracy of 67.5%, in conjunction with their development and updated counterparts.

Table 3: The factual accuracy of different LLMs on single-hop questions.

LLM	Question Type	Hum.	Soc. Sci.	Stem	Ave	Summary
GPT-3.5	Yes-No	72.2	74.1	73.9	73.4	60.1
	MC	68.6	67.0	60.5	65.4	
	WH	45.3	44.3	35.3	41.6	
GPT-4o	Yes-No	84.4	82.1	79.4	82.0	69.0
	MC	82.9	79.2	72.6	78.2	
	WH	50.8	51.4	38.4	46.9	
GPT-4	Yes-No	81.0	80.7	81.0	80.9	67.5
	MC	79.2	77.6	70.9	75.9	
	WH	49.1	48.4	39.3	45.6	
DeepSeek-V3	Yes-No	76.2	75.6	77.2	76.3	61.1
	MC	65.0	65.0	55.5	61.8	
	WH	51.7	44.9	39.0	45.2	
Claude-3.5-Haiku	Yes-No	76.9	78.9	76.5	77.4	62.4
	MC	71.2	69.9	62.9	68.0	
	WH	45.5	45.4	34.1	41.7	
Claude-Sonnet-4.0	Yes-No	78.1	79.0	78.9	78.7	66.5
	MC	75.3	74.3	65.2	71.6	
	WH	52.8	50.2	44.3	49.1	
Gemini-2.0	Yes-No	81.3	78.9	76.0	78.7	66.2
	MC	77.7	75.6	69.6	74.3	
	WH	48.8	50.3	37.7	45.6	
Qwen-3	Yes-No	74.4	72.9	74.5	73.9	59.9
	MC	67.4	66.0	64.9	66.1	
	WH	42.0	41.2	35.7	39.6	
Qwen-3-Reasoning	Yes-No	72.8	75.1	76.6	74.8	58.3
	MC	63.9	62.6	61.2	62.6	
	WH	41.8	37.2	33.1	37.4	

WH questions are much harder for LLMs. Comparing the results across different question types, all LLMs exhibit the lowest performance on WH questions, with an average accuracy of 37.4%, suggesting that this particular question type poses a considerable challenge for LLMs.

Multi-hop questions are more challenging for LLMs. In addition to single-hop questions, FactChecker has the capability to generate multi-hop questions. To assess the effectiveness of multi-hop questions, we employ FactChecker to generate 1000 such questions per domain and per question type, utilizing them to query all LLMs, subsequently evaluating the accuracy of their responses. As demonstrated in Table 4, it is evident that all LLMs experience a higher incidence of factual errors when faced with 2-4 hop questions, in comparison to single-hop questions.

Moreover, The results reveal a decreasing trend in factual accuracy as the number of hops increases for all LLMs. Notably, the decline is steeper from 1-hop to 2-hop questions, with a more gradual slowdown in the drop as we move from 2-hop to 4-hop questions. This indicates that while multi-hop reasoning introduces greater complexity and leads to more factual errors, the rate of decline in accuracy becomes less pronounced with increasing hops.

3.3 Effectiveness of FactProbe

This section investigates the effectiveness of FactProbe in enhancing the detection of factual errors and providing deeper insights into LLM limitations, building on FactChecker's established capabilities.

FactProbe amplifies the detection of factual

Table 4: The factual accuracy of different LLMs on multi-hop questions.

LLM	Hop			
	1-hop	2-hop	3-hop	4-hop
GPT-3.5	69.4	60.2	46.5	44.8
GPT-4o	80.1	66.6	54.2	53.6
GPT-4	78.4	65.9	52.8	52.2
DeepSeek-V3	69.1	54.1	39.1	37.5
Claude-3.5-Haiku	72.7	55.0	47.0	44.3
Claude-Sonnet-4.0	75.1	62.0	43.9	36.1
Gemini-2.0	76.5	64.2	51.8	49.6
Qwen-3	70.0	59.6	48.7	46.2
Qwen-3-reasoning	68.7	56.6	45.8	43.1

errors across LLMs. Leveraging FactChecker's foundation, FactProbe iteratively refines question sets based on initial LLM responses, targeting identified weaknesses to uncover a higher incidence of factual errors. Starting with 1000 questions per topic, subsequent trials show a marked decline in accuracy, as seen in Tables 8. For instance, GPT-4o's accuracy drops from 84.4% to 65.8% on Yes-No questions and from 82.9% to 54.1% on Multiple-Choice questions over five trials, demonstrating FactProbe's ability to progressively expose vulnerabilities of LLMs.

FactProbe exacerbates challenges with WH and multi-hop questions. Extending FactChecker's finding that WH questions (average accuracy 37.4%) and multi-hop questions pose significant difficulties, FactProbe intensifies these challenges. Across trials, WH question accuracy plummets (e.g., GPT-3.5 from 45.3% to 7.5%), while in Table 9, multi-hop accuracies decline sharply than single-hop (e.g., GPT3.5 from 64.2% to 26.4% on 2-hop MC questions), underscoring FactProbe's effectiveness in probing complex reasoning weaknesses.

3.4 Validity of Identified Factual Errors

In this section, we investigate whether the factual errors exposed by FactChecker are true failures through manual inspection. We manually inspect the 100 failure cases to study their validity. Specifically, we recruit three annotators, with Bachelor's degrees or above and proficiency in English, to answer the questions manually with the help of the Internet, then discuss their answers to resolve the disagreement, and finally annotate each failure case as a valid error or false negative. The result shows that among 100 randomly generated cases, 93 cases are valid errors, indicating that the identified factual errors are reliable.

4 Related Work

The prevalent approach of factual evaluation involves reference-based techniques, which rely on benchmarks created through manual question design or test input labeling (Clark et al., 2019; McCoy et al., 2019; Lin et al., 2021; Talmor et al., 2019b; Laskar et al., 2023). Despite their utility, these methods demand substantial human effort and are dependent on the development of comprehensive benchmarks. Lewis et al. (2021) is a pioneering work that automatically generates massive question-answer pairs to evaluate open-domain question-answering models. Recently, Sun et al. (2023) and Omar et al. (2023) also adopted knowledge graphs to evaluate the factual correctness of LLMs. However, their scope of question types and topics, testbed models, and evaluation methods are limited, compared with FactChecker.

Additionally, static benchmarks are prone to data contamination, posing challenges in accurately evaluating LLMs and efficiently identifying errors. Recent advancements have seen the emergence of automatic test case generation, independent of manually pre-annotated labels (Chen et al., 2021; Liu et al., 2022; Shen et al., 2022). However, these techniques still depend on existing benchmarks for question formulation, limiting the breadth of test case generation.

To sum up, our approach differs significantly in several key aspects: (1) **Scope**: FactChecker offers a more extensive scope, capable of generating three types of questions on any topic, as opposed to their method which is restricted to probing specific relationships, like "place of birth" or "date of birth", in a cloze-style format. (2) **Testbed**: FactChecker evaluates six leading LLMs, while most of the previous works focused solely on the traditional models. (3) **Testing Logic**: FactChecker is an automated testing framework designed to dynamically generate a diverse array of test cases each time it is run, aiming to avoid data contamination.

5 Conclusion

In this paper, we design and implement FactChecker, an automated framework dedicated to uncovering factual errors in LLMs. Distinct from previous approaches that depend on extensive human annotation or are prone to data contamination, FactChecker leverages a structured KG to autonomously generate a wide array of questions spanning various topics and relations. We conducted comprehensive evaluations using six prominent models. Our empirical findings reveal that FactChecker can successfully identify factual errors and enhance the factual accuracy of LLMs.

Limitations

This paper has two primary limitations that offer avenues for future research:

- The effectiveness of FactChecker is affected by the quality of the knowledge graph (*e.g.*Wikidata), which is hard to guarantee. Please refer to our discussion in Appendix A.
- This paper does not provide any novel method to improve the factual correctness of LLMs. More effective and efficient methods are needed to further enhance the factual correctness of LLMs.

References

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on chatgpt? *ArXiv*, abs/2303.12767.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, 22(82):1–6.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023.

Songqiang Chen, Shuo Jin, and Xiaoyuan Xie. 2021. Testing your question answering software via asking recursively. *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 104–116.

Yulong Chen, Yang Liu, Jianhao Yan, Xuefeng Bai, Ming Zhong, Yinghao Yang, Ziyi Yang, Chenguang Zhu, and Yue Zhang. 2024. See what llms cannot answer: A self-challenge framework for uncovering llm weaknesses. *Preprint*, arXiv:2408.08978.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *North American Chapter of the Association for Computational Linguistics*.

Yue Zhang Cunxiang Wang, Pai. 2021. Can generative pre-trained language models serve as knowledge bases for closed-book QA? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Shuzheng Gao, Xinjie Wen, Cuiyun Gao, Wenxuan Wang, and Michael R. Lyu. 2023. Constructing effective in-context demonstration for code intelligence tasks: An empirical study. *ArXiv*, abs/2304.07575.

Cindy Gordon. 2023. Chatgpt is the fastest growing app in the history of web applications. `https://www.forbes.com/sites/cindygordon/2023/02/02`. Accessed: 2023-07-01.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine. *arXiv preprint arXiv:2301.08745*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual lama: Investigating knowledge in multilingual pretrained language models. *EACL*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq R. Joty, and J. Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. In *Annual Meeting of the Association for Computational Linguistics*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Kuttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Wangyue Li, Liangzhi Li, Tong Xiang, Xiao Liu, Wei Deng, and Noa Garcia. 2024. Can multiple-choice questions really be useful in detecting the abilities of llms? *ArXiv*, abs/2403.17752.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics*.

Zixi Liu, Yang Feng, Yining Yin, J. Sun, Zhenyu Chen, and Baowen Xu. 2022. Qatest: A uniform fuzzing framework for question answering systems. *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *ACL*.

Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *ArXiv*, abs/2302.06466.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *ArXiv*, abs/1909.01066.

Daw Khin Po. 2020. Similarity based information retrieval using levenshtein distance algorithm. *International Journal of Advances in Scientific Research and Engineering*.

Qingchao Shen, Junjie Chen, J Zhang, Haoyu Wang, Shuang Liu, and Menghan Tian. 2022. Natural test generation for precise testing of question answering software. *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*.

Kai Sun, Y. Xu, Hanwen Zha, Yue Liu, and Xinhsuai Dong. 2023. Head-to-tail: How knowledgeable are large language models (llms)? a.k.a. will llms replace knowledge graphs? *ArXiv*, abs/2308.10168.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019a. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019b. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *NAACL*.

Jen tse Huang, Jianping Zhang, Wenxuan Wang, Pinjia He, Yuxin Su, and Michael R. Lyu. 2022. Aeon: a method for automatic evaluation of nlp test cases. *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*.

Hao Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael R. Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *ArXiv*, abs/2303.13648.

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. *Preprint*, arXiv:1809.09600.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. *Preprint*, arXiv:1904.10281.

A Threats to Validity

The validity of this work may be subject to several potential threats.

The first concern is the reliance on NLP techniques employed by FactChecker for error detection. Given the inherent limitations of NLP methods, FactChecker might generate false positives or overlook errors, resulting in false negatives. This is particularly evident in scenarios where varying interpretations of correct responses to WH questions challenge accurate validation. To mitigate this issue, we evaluated the efficacy of several prominent similarity methods, selecting the most effective one based on performance metrics. Additionally, we conducted human annotation to demonstrate that FactChecker achieves high accuracy in error detection, as evidenced by the results.

The second threat is from the implementation of FactChecker, which covers only one knowledge base, Wikidata. Like any knowledge base, Wikidata is prone to factual inaccuracies or suffers from incomplete data, leading to sub-optimal question generation. Additionally, it is vulnerable to issues like data leakage. To address these two concerns, we adopt strategies respectively: (1) FactChecker is designed for flexibility, allowing easy substitution of Wikidata with alternative knowledge bases. Incorporating multiple knowledge bases can enhance the robustness and quality of the generated questions. (2) One advantage of Wikidata is the graph format for information storage, a method not extensively employed in training most LLMs despite its public availability. Our primary contribution lies in the development of an automated testing framework. This framework aims to minimize the human effort needed to identify factual inaccuracies within LLMs. Essentially, FactChecker flags potential errors, which are then subjected to further human analysis to assess their validity.

The third limitation of our study is the limited exploration of various LLMs during evaluation. Our current analysis does not encompass a broad assessment of FactChecker's performance across numerous systems. To address this limitation, we focus on testing the most prevalent conversational LLMs and SOTA academic models developed by major corporations. Future work, utilizing FactChecker, could expand this scope to include additional commercial and research models, thereby enhancing the robustness of our findings.

B Illustration of The Retrieval Process of Fact Triplets.

A fact triplet is represented in the form of (SUBJECT, relation, OBJECT). For instance, the triplet (USA, capital, Washington D.C.) denotes the fact that the capital of the USA is Washington D.C. FactChecker enables users to obtain fact triplets pertaining to specific topics. As illustrated in Figure 2, when a user expresses interest in the topic of emperors, FactChecker proceeds to convert the "occupation: emperor" specification into a SPARQL query language[6], which is utilized for querying related triplets in Wikidata. The resulting SPARQL query will retrieve all accessible fact triplets about emperors, including examples such as "Napoleon, place of birth, Ajaccio" and "Peter the Great, father, Alexei I of Russia".

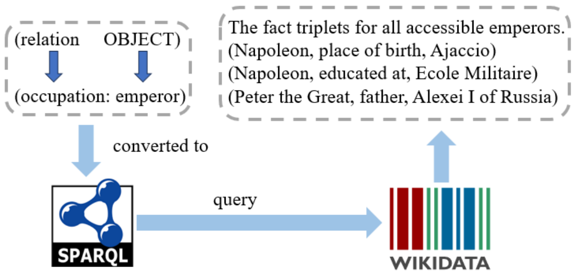
(relation OBJECT)
(occupation: emperor)
converted to
query
SPARQL
WIKIDATA
The fact triplets for all accessible emperors.
(Napoleon, place of birth, Ajaccio)
(Napoleon, educated at, Ecole Militaire)
(Peter the Great, father, Alexei I of Russia)

Figure 2: The retrieval process for fact triplets.

C Illustration of the Rule-Based Method for Question Generation

FactChecker utilizes a rule-based approach to generate questions from the constructed KG. It can generate various types of questions, including different question types, *i.e.*, Yes-No questions, MC questions and WH questions. For each triplet in the constructed knowledge graph, FactChecker converts it to the question form according to their POS and NER features. Figure 3 shows some examples of FactChecker's question generation method.

D Details of Selected Topics

To comprehensively evaluate LLMs' performance, we conduct experiments by generating questions from three domains: Humanity, Social Science and Science, technology, engineering, and mathematics (STEM), and each domain consists of not less than three topics, the details and examples of which are shown in Table 5

E Preliminary Experiments

In this section, we conducted an initial experiment to validate our choice of employing a rule-based method for question generation as opposed to directly instructing ChatGPT to craft questions from fact triplets. Additionally, we conducted experiments to assess the effectiveness of our grammar-

[6]https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service

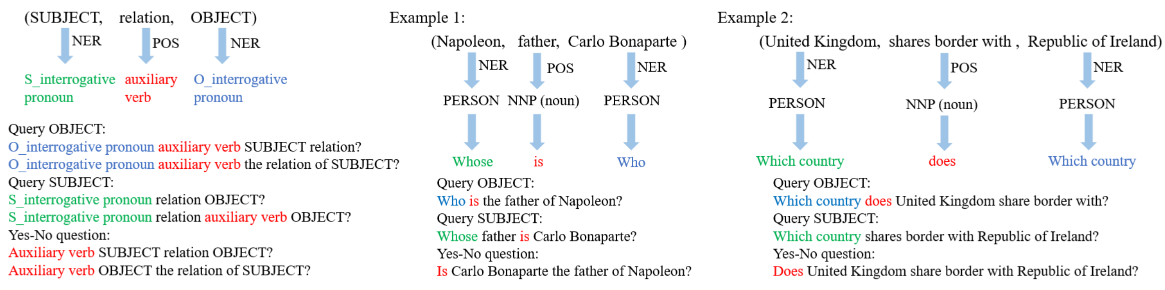
(SUBJECT, relation, OBJECT)
NER
POS
NER
S_interrogative pronoun
auxiliary verb
O_interrogative pronoun
Query OBJECT:
O_interrogative pronoun auxiliary verb SUBJECT relation?
O_interrogative pronoun auxiliary verb the relation of SUBJECT?
Query SUBJECT:
S_interrogative pronoun relation OBJECT?
S_interrogative pronoun relation auxiliary verb OBJECT?
Yes-No question:
Auxiliary verb SUBJECT relation OBJECT?
Auxiliary verb OBJECT the relation of SUBJECT?
Example 1:
(Napoleon, father, Carlo Bonaparte)
NER
POS
NER
PERSON
NNP (noun)
PERSON
Whose
is
Who
Query OBJECT:
Who is the father of Napoleon?
Query SUBJECT:
Whose father is Carlo Bonaparte?
Yes-No question:
Is Carlo Bonaparte the father of Napoleon?
Example 2:
(United Kingdom, shares border with , Republic of Ireland)
NER
POS
NER
PERSON
NNP (noun)
PERSON
Which country
does
Which country
Query OBJECT:
Which country does United Kingdom share border with?
Query SUBJECT:
Which country shares border with Republic of Ireland?
Yes-No question:
Does United Kingdom share border with Republic of Ireland?

Figure 3: The proposed rule-based method for Question Generation.

Table 5: Selected topics for evaluation.

Domain	Topic	Example
Hum.	Art	Monna Lisa
	History	World War II
	Philosophy	Plato
Soc. Sci	Psychology	Sigmund Freud
	Sociology	Elitism
	Geography	Earthquake
STEM	Mathematics	Pythagorean theorem
	Physics	Planck constant
	Computer Science	Dijkstra's algorithm

checking and polishing modules. We also meticulously investigated the comparison of five evaluation metrics.

E.1 Can ChatGPT outperform the proposed rule-based methods on question generation?

Given the capabilities of ChatGPT, an alternative approach for generating questions from fact triplets involves instructing ChatGPT to generate the desired questions based on the extracted fact triplets. To verify the viability of this approach, we prompted ChatGPT to generate 200 questions from fact triplets and compared the results with the rule-based method we proposed. Subsequently, we enlisted the assistance of three annotators, each holding a Bachelor's degree or higher and proficient in English, to independently evaluate the quality of the generated questions. Any discrepancies in their assessments were resolved through discussion. The results indicate that while ChatGPT is capable of producing some high-quality questions from fact triplets, it may occasionally deviate from our instructions, introducing unreliability. Among the 200 questions generated, the annotators found that 26 did not align with our expectations. On the other hand, despite introducing some grammatical errors, the rule-based method produced questions where 98.5% adhered to the intended semantic meaning.

E.2 Are the modules of grammar-checking and rephrasing effective?

In order to address potential grammatical errors introduced by rule-based approaches in question generation, we have implemented and compared two modules within our method. The first approach incorporates the use of a grammar checker API to filter out questions exhibiting grammatical errors. Following this filtering process, the remaining questions maintain a high level of quality. However, this approach has a drawback as it tends to be overly sensitive, leading to the elimination of approximately 50% of all generated questions, resulting in a notable false positive rate. The second alternative entails instructing ChatGPT to paraphrase the generated questions, thereby rectifying grammatical errors and enhancing the natural sound of the questions. Our hired annotators observed that by directly leveraging ChatGPT for paraphrasing, the question formats became more diverse, and all 48 questions initially containing grammar errors were successfully corrected. The final results demonstrate that relying solely on the rewriting approach yields better overall performance. Thus, FactChecker adopts the rewriting powered by ChatGPT to obtain fluent test cases.

E.3 Which similarity metric performs the best?

Due to the diverse nature of the responses generated for WH questions, utilizing a straightforward exact match criterion is not sufficient for addressing these variations effectively. As a result, we compare five distinct evaluation methods described in Section 2.3.2. The objective is to identify the most effective method that yields satisfactory results. To conduct the evaluation, we randomly selected 500 questions along with their corresponding generated responses from LLMs and the ground-truth answers. Subsequently, the recruited three annotators are required to annotate whether the generated response matches the ground-truth answers. Finally, we obtained 238 cases that the responses are annotated as not aligned with the ground truth. Then, we

Table 6: Performance of different evaluation methods.

Evaluation Method	Precision	Recall	F1
Levenshtein distance	72.6	99.2	83.8
N-grams	61.7	100	76.3
Word embedding	72.8	91.2	81.0
Sentence transformer	78.2	97.9	87.0
ChatGPT	100	65.5	79.2

use the annotated 500 data as a benchmark to evaluate the performance of the five matching methods. The results are shown in Table 6, demonstrating that the sentence transformer method exhibits the most promising performance, with the highest F1 score. In other words, the sentence transformer can successfully identify nearly all the incorrect responses while maintaining a smaller number of false positive cases. Thus, FactChecker adopts the sentence transformer as the matching metrics for WH questions.

F Comprehensive Evaluation Results

This section evaluates FactChecker's effectiveness in measuring the factual accuracy of Large Language Models (LLMs) across single-hop, 2-hop, and multi-hop questions. We begin with performance trends for multi-hop questions, detailed by domain and question type, followed by results for single-hop and 2-hop questions. We also highlight FactProbe's role in generating and refining questions for single-hop and 2-hop scenarios, demonstrating its ability to uncover LLM inaccuracies.

F.1 FactChecker's Multi-Hop Question Performance

Table 7 presents the factual accuracy of LLMs on multi-hop questions (1-hop to 4-hop) across three domains—Humanities (Hum.), Social Sciences (Soc. Sci.), and STEM—and two question types (Yes-No and Multiple-Choice). Each trial includes 1,000 unique questions per domain and question type, with no duplicates within the same category. The results show a consistent decline in accuracy as the number of hops increases, with Multiple-Choice questions proving more challenging due to distractors, especially in STEM domains.

F.2 FactProbe's Single-Hop Question Performance

Table 8 reports the factual accuracy of LLMs on single-hop questions across three domains (Humanities, Social Sciences, and STEM) and three question types (Yes-No, Multiple-Choice, and WH). Each trial consists of 1,000 unique questions per domain and question type, ensuring no duplicates within the same category. FactProbe 's iterative refinement targets error patterns, reducing accuracy in later trials and revealing persistent factual weaknesses across LLMs.

F.3 FactProbe's 2-Hop Question Performance

Table 9 reports the factual accuracy of LLMs on 2-hop questions across three domains (Humanities, Social Sciences, and STEM) and three question types (Yes-No, Multiple-Choice, and WH). Each trial includes 1,000 unique questions per domain and question type, with no duplicates within the same category. FactProbe 's adaptive question generation, which focuses on error-prone areas identified in initial trials, significantly reduces accuracy in subsequent trials, underscoring LLMs' challenges with sequential reasoning tasks.

Table 7: The factual accuracy of different LLMs on multi-hop questions.

LLM	Domain	Question Type	Hop			
			1-hop	2-hop	3-hop	4-hop
GPT-3.5	Hum.	Yes/No	72.2	68.7	54.9	51.4
		MC	68.6	64.2	46.0	42.4
	Soc. Sci.	Yes/No	74.1	66.2	53.9	50.0
		MC	67.0	50.3	39.2	43.8
	STEM	Yes/No	73.9	66.4	53.8	51.4
		MC	60.5	45.2	31.3	29.5
GPT-4o	Hum.	Yes/No	84.4	74.4	62.9	60.6
		MC	82.9	74.8	56.7	50.6
	Soc. Sci.	Yes/No	82.1	68.8	59.7	64.9
		MC	79.2	61.5	54.1	55.4
	STEM	Yes/No	79.4	70.5	57.0	56.4
		MC	72.6	49.6	34.6	33.4
GPT-4	Hum.	Yes/No	81.0	73.4	61.2	60.9
		MC	79.2	72.4	55.0	47.7
	Soc. Sci.	Yes/No	80.7	68.3	58.7	65.4
		MC	77.6	62.0	53.0	53.8
	STEM	Yes/No	81.0	69.2	56.5	53.9
		MC	70.9	50.2	32.3	31.3
DeepSeek-V3	Hum.	Yes/No	76.2	67.7	56.0	52.1
		MC	65.0	56.3	31.5	28.0
	Soc. Sci.	Yes/No	75.6	66.2	55.6	51.0
		MC	65.0	45.2	24.5	26.5
	STEM	Yes/No	77.2	62.4	53.0	51.5
		MC	55.5	26.9	14.1	16.0
Claude-3.5-Haiku	Hum.	Yes/No	76.9	70.5	60.4	55.6
		MC	71.2	55.7	39.7	33.6
	Soc. Sci.	Yes/No	78.9	55.4	60.0	60.0
		MC	69.9	43.8	36.0	34.9
	STEM	Yes/No	76.5	68.1	56.8	54.9
		MC	62.9	36.4	29.3	27.0
Claude-Sonnet-4.0	Hum.	Yes/No	78.1	70.2	60.1	52.7
		MC	75.3	68.5	33.4	19.3
	Soc. Sci.	Yes/No	79.0	70.2	59.9	55.9
		MC	74.3	56.8	31.4	22.6
	STEM	Yes/No	78.9	66.1	54.8	51.6
		MC	65.2	40.2	23.5	14.2
Gemini-2.0	Hum.	Yes/No	81.3	73.1	62.0	57.8
		MC	77.7	70.1	52.1	45.9
	Soc. Sci.	Yes/No	78.9	67.6	57.4	56.2
		MC	75.6	59.4	49.1	51.0
	STEM	Yes/No	76.0	68.5	55.1	54.4
		MC	69.6	46.5	35.2	32.1
Qwen-3	Hum.	Yes/No	74.4	69.3	59.0	55.6
		MC	67.4	63.4	45.1	42.0
	Soc. Sci.	Yes/No	72.9	65.5	59.8	56.2
		MC	66.0	51.6	43.4	43.1
	STEM	Yes/No	74.5	65.0	53.9	52.1
		MC	64.9	42.5	31.0	28.4
Qwen-3-Reasoning	Hum.	Yes/No	72.8	67.6	56.1	54.1
		MC	63.9	59.9	41.0	34.5
	Soc. Sci.	Yes/No	75.1	65.8	57.6	52.0
		MC	62.6	44.0	37.3	38.3
	STEM	Yes/No	76.6	61.2	52.9	51.6
		MC	61.2	41.3	30.1	28.3

G Full Text of Human Annotation Instruction

Here are the full text of instructions given to participants in section E.1.

Thank you for participating in this evaluation. Your task is to assess 200 questions generated from fact triplets to determine if they accurately reflect the triplet's semantic meaning. For each assigned question-triplet pair, review the triplet and question, then record a Yes (question aligns with the triplet's content and intent) or No (question misrepresents the triplet or deviates from instructions) in the provided options. Noted that all data will only be used as research purpose.

H Ethical Statement

All participants were informed beforehand that their feedback would be used for research purposes and potentially published in this paper. No personal or private information was collected, and all data collected was anonymized to ensure privacy. Participation was voluntary, and the recruited volunteers were proficient in using LLMs to ensure informed contributions.

Table 8: Factual accuracy of LLMs across multiple trials on single-hop questions.

LLM	Trial	Hum.			Soc. Sci.			STEM		
		Yes/No	MC	WH	Yes/No	MC	WH	Yes/No	MC	WH
GPT-3.5	Seed	72.2	68.6	45.3	74.1	67.0	44.3	73.9	60.5	35.3
	Trial 1	61.8	52.6	18.5	58.1	50.4	23.4	62.1	48.1	18.4
	Trial 2	57.5	49.6	12.5	52.8	44.2	21.1	55.1	44.4	15.8
	Trial 3	56.2	46.9	11.4	53.0	42.4	15.4	57.2	44.6	15.5
	Trial 4	55.3	45.5	9.8	51.9	39.9	14.4	55.4	42.4	11.1
	Trial 5	53.6	46.2	7.5	48.1	38.2	12.1	53.2	37.3	11.1
GPT-4o	Seed	84.4	82.9	50.8	82.1	79.2	51.4	79.4	72.6	38.4
	Trial 1	71.0	68.3	20.4	74.5	65.2	21.4	71.0	61.3	21.1
	Trial 2	66.2	58.9	15.0	68.1	54.1	16.5	65.7	57.4	15.2
	Trial 3	67.3	56.4	12.0	67.4	54.7	14.9	65.7	54.9	15.1
	Trial 4	65.7	51.7	8.8	63.2	50.8	11.4	61.0	52.0	10.8
	Trial 5	65.8	54.1	9.1	63.6	51.5	10.6	61.2	50.5	10.3
GPT-4	Seed	81.0	79.2	49.1	80.7	77.6	48.4	81.0	70.9	39.3
	Trial 1	69.0	67.1	25.4	63.4	57.2	24.2	61.9	54.1	18.4
	Trial 2	59.4	61.6	18.4	65.1	56.5	21.8	60.8	50.1	16.8
	Trial 3	61.3	61.1	16.0	59.2	49.6	20.8	61.9	48.2	14.6
	Trial 4	62.3	60.8	14.3	60.2	53.1	15.2	57.4	46.2	13.0
	Trial 5	61.0	57.5	14.4	57.4	49.1	14.5	55.0	45.8	9.8
DeepSeek-V3	Seed	76.2	65.0	51.7	75.6	65.0	44.9	77.2	55.5	39.0
	Trial 1	63.1	52.3	23.6	61.8	43.4	20.1	59.7	37.5	20.0
	Trial 2	57.8	49.9	16.7	56.6	43.8	18.8	58.8	32.1	14.2
	Trial 3	52.8	49.2	11.3	58.3	41.5	17.1	55.0	28.4	13.9
	Trial 4	55.6	45.9	11.5	50.5	38.0	14.4	50.8	30.0	12.2
	Trial 5	52.3	43.2	10.1	51.6	37.9	10.9	51.1	29.6	10.8
Claude-3.5-Haiku	Seed	76.9	71.2	45.5	78.9	69.9	45.4	76.5	62.9	34.1
	Trial 1	66.4	54.2	17.9	65.2	47.0	22.8	60.7	41.3	18.7
	Trial 2	62.1	49.1	15.3	60.6	42.7	18.6	57.4	40.0	15.9
	Trial 3	61.1	48.1	13.7	60.1	40.5	19.0	58.3	38.1	15.3
	Trial 4	57.4	45.4	11.8	59.4	37.0	16.1	54.4	34.7	13.0
	Trial 5	57.6	44.9	11.5	57.8	38.2	16.3	56.1	36.4	12.3
Claude-Sonnet-4.0	Seed	78.1	75.3	52.8	79.0	74.3	50.2	78.9	65.2	44.3
	Trial 1	64.8	66.4	23.1	58.3	59.3	25.8	56.7	45.6	21.6
	Trial 2	63.0	62.8	16.2	52.7	56.7	21.6	56.6	34.4	15.9
	Trial 3	60.7	59.5	14.6	49.8	55.4	21.6	52.5	30.8	16.7
	Trial 4	61.0	58.6	14.0	48.6	47.6	17.4	54.4	26.0	13.5
	Trial 5	55.0	57.1	10.4	44.9	42.5	16.0	53.1	24.7	13.4
Gemini-2.0	Seed	81.3	77.7	48.8	78.9	75.6	50.3	76.0	69.6	37.7
	Trial 1	65.7	63.4	21.2	62.8	64.4	24.3	61.3	52.0	20.1
	Trial 2	63.5	61.1	15.3	63.4	62.1	23.3	59.9	47.7	16.1
	Trial 3	62.7	60.5	12.0	62.4	57.8	22.2	59.0	43.5	14.7
	Trial 4	60.9	60.2	12.4	60.5	58.2	20.3	55.6	42.6	15.9
	Trial 5	62.4	55.2	11.6	59.7	50.0	23.7	56.5	41.6	11.3
Qwen-3	Seed	74.4	67.4	42.0	72.9	66.0	41.2	74.5	64.9	35.7
	Trial 1	66.8	52.3	18.9	62.3	52.4	18.1	63.2	47.8	16.2
	Trial 2	61.9	51.6	18.0	57.3	46.2	18.9	60.3	42.8	18.1
	Trial 3	60.1	47.7	14.9	57.9	43.9	14.1	58.1	40.6	13.2
	Trial 4	58.6	49.5	12.4	53.3	40.5	15.2	55.2	41.5	13.5
	Trial 5	59.0	42.6	13.8	55.3	38.4	14.1	54.4	39.1	13.8
Qwen-3-Reasoning	Seed	72.8	63.9	41.8	75.1	62.6	37.2	76.6	61.2	33.1
	Trial 1	58.8	51.1	18.5	60.9	47.2	15.3	60.6	42.1	15.6
	Trial 2	57.3	49.1	15.1	58.9	45.5	14.9	54.0	42.2	11.5
	Trial 3	56.3	45.7	14.7	54.6	42.1	12.9	58.9	37.4	13.8
	Trial 4	58.7	45.7	12.5	52.1	40.5	12.0	54.0	35.7	12.6
	Trial 5	55.7	44.5	12.2	53.9	37.1	11.6	52.5	35.6	10.7

Table 9: The factual accuracy of LLMs across multiple trials on 2-hop questions.

LLM	Trial	Hum.		Soc. Sci.		STEM	
		Yes/No	MC	Yes/No	MC	Yes/No	MC
GPT-3.5	Seed	68.7	64.2	66.2	50.3	66.4	45.2
	Trial 1	54.5	47.5	54.8	37.7	53.7	34.0
	Trial 2	52.4	36.5	54.0	35.1	51.2	33.1
	Trial 3	52.3	30.1	53.0	32.5	50.4	29.0
	Trial 4	51.4	27.8	51.9	30.6	48.3	26.2
	Trial 5	51.1	26.4	51.0	29.4	44.7	23.8
GPT-4o	Seed	74.4	74.8	68.8	61.5	70.5	49.6
	Trial 1	62.6	52.1	56.9	44.5	58.7	42.3
	Trial 2	60.4	45.9	55.2	40.4	56.0	35.5
	Trial 3	58.5	41.5	54.5	37.5	54.4	32.5
	Trial 4	57.5	38.0	53.3	34.8	52.3	30.1
	Trial 5	57.1	35.6	53.0	33.1	51.7	28.3
GPT-4	Seed	73.4	72.4	68.3	62.0	69.2	50.2
	Trial 1	62.2	55.7	57.9	49.8	54.8	43.7
	Trial 2	58.4	47.2	59.0	54.2	53.7	37.3
	Trial 3	56.4	41.5	56.8	53.8	53.0	34.7
	Trial 4	56.4	38.2	55.3	53.4	51.9	32.1
	Trial 5	55.8	35.9	54.7	52.9	51.1	30.3
DeepSeek-V3	Seed	67.7	56.3	66.2	45.2	62.4	26.9
	Trial 1	54.7	48.6	53.9	38.4	54.2	26.1
	Trial 2	53.2	44.4	51.9	35.3	51.6	36.8
	Trial 3	52.1	38.8	51.1	32.4	51.3	35.4
	Trial 4	51.7	35.5	50.5	30.8	48.7	34.2
	Trial 5	51.5	33.2	50.0	28.7	47.2	33.1
Claude-3.5-Haiku	Seed	70.5	55.7	55.4	43.8	68.1	36.4
	Trial 1	59.5	38.8	54.9	40.6	57.7	37.3
	Trial 2	56.7	38.3	54.3	37.9	54.1	32.3
	Trial 3	56.0	34.5	53.5	35.1	53.7	30.0
	Trial 4	54.7	32.8	53.1	32.5	52.0	28.9
	Trial 5	53.6	31.8	52.6	31.6	50.2	28.0
Claude-Sonnet-4.0	Seed	70.2	68.5	70.2	56.8	66.1	40.2
	Trial 1	55.7	48.2	54.5	40.5	52.0	39.9
	Trial 2	54.4	36.6	53.8	30.8	50.2	34.8
	Trial 3	54.4	31.0	53.6	25.2	49.1	31.0
	Trial 4	53.5	26.8	52.9	21.9	47.9	28.2
	Trial 5	52.6	24.0	52.5	19.7	47.7	26.2
Gemini-2.0	Seed	73.1	70.1	67.6	59.4	68.5	46.5
	Trial 1	60.3	50.6	52.8	42.9	52.4	41.8
	Trial 2	58.4	45.3	51.6	37.3	50.9	37.4
	Trial 3	56.5	40.4	51.7	34.7	50.5	33.8
	Trial 4	55.9	37.4	50.7	32.3	49.7	30.3
	Trial 5	55.3	35.1	50.5	30.5	48.9	27.8
Qwen-3	Seed	69.3	63.4	65.5	51.6	65.0	42.5
	Trial 1	59.3	49.3	54.5	43.8	53.5	39.8
	Trial 2	55.6	36.8	52.7	36.0	43.9	23.6
	Trial 3	54.8	31.0	53.2	31.1	33.2	23.5
	Trial 4	54.0	30.6	50.0	28.7	24.2	18.5
	Trial 5	53.7	30.5	47.3	26.3	42.8	16.9
Qwen-3-Reasoning	Seed	67.6	59.9	65.8	44.0	61.2	41.3
	Trial 1	56.8	44.5	53.3	39.5	55.5	42.5
	Trial 2	57.1	35.7	52.2	29.5	49.4	26.2
	Trial 3	54.5	31.3	51.8	29.4	52.2	25.9
	Trial 4	52.9	31.5	51.2	25.9	45.5	20.2
	Trial 5	54.9	25.6	51.0	25.5	43.9	18.3