OntoFact: Unveiling Fantastic Fact-Skeleton of LLMs via Ontology-Driven Reinforcement Learning

Ziyu Shang[1*], **Wenjun Ke**[1,2*†], **Nana Xiu**[3], **Peng Wang**[1,2†],
Jiajun Liu[1], **Yanhui Li**[4], **Zhizhao Luo**[5], **Ke Ji**[1]

[1]School of Computer Science and Engineering, Southeast University
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications
(Southeast University), Ministry of Education, China
[3]School of Cyber Science and Engineering, Southeast University
[4]State Key Laboratory for Novel Software Technology, Nanjing University
[5]Beijing Institute of Computer Technology and Application
{ziyus1999, kewenjun, xiunana, pwang, jiajliu, keji}@seu.edu.cn, yanhuili@nju.edu.cn, qibai-aluminum@outlook.com

Abstract

Large language models (LLMs) have demonstrated impressive proficiency in information retrieval, while they are prone to generating incorrect responses that conflict with reality, a phenomenon known as intrinsic hallucination. The critical challenge lies in the unclear and unreliable fact distribution within LLMs trained on vast amounts of data. The prevalent approach frames the factual detection task as a question-answering paradigm, where the LLMs are asked about factual knowledge and examined for correctness. However, existing studies primarily focused on deriving test cases only from several specific domains, such as movies and sports, limiting the comprehensive observation of missing knowledge and the analysis of unexpected hallucinations. To address this issue, we propose OntoFact, an adaptive framework for detecting unknown facts of LLMs, devoted to mining the ontology-level skeleton of the missing knowledge. Specifically, we argue that LLMs could expose the ontology-based similarity among missing facts and introduce five representative knowledge graphs (KGs) as benchmarks. We further devise a sophisticated ontology-driven reinforcement learning (ORL) mechanism to produce error-prone test cases with specific entities and relations automatically. The ORL mechanism rewards the KGs for navigating toward a feasible direction for unveiling factual errors. Moreover, empirical efforts demonstrate that dominant LLMs are biased towards answering `Yes` rather than `No`, regardless of whether this knowledge is included. To mitigate the overconfidence of LLMs, we leverage a hallucination-free detection (HFD) strategy to tackle unfair comparisons between baselines, thereby boosting the result robustness. Experimental results on 5 datasets, using 32 representative LLMs, reveal a general lack of fact in current LLMs. Notably, ChatGPT exhibits fact error rates of 51.6% on DBpedia and 64.7% on YAGO, respectively. Additionally, the ORL mechanism demonstrates promising error prediction scores, with F1 scores ranging from 70% to 90% across most LLMs. Compared to the exhaustive testing, ORL achieves an average recall of 80% while reducing evaluation time by 35.29% to 63.12%.

Introduction

Large language models (LLMs) have proven remarkable effectiveness across various NLP tasks (Bubeck et al. 2023; Li, Wang, and Ke 2023). These models, trained on massive corpora, encode world knowledge within enormous parameters, making them adaptable to knowledge-intensive tasks (Liévin, Hother, and Winther 2022; Singhal et al. 2023). However, LLMs often lack factual accuracy, which is crucial for applications where credibility is paramount (Maynez et al. 2020; Kang and Hashimoto 2020). Understanding the factual distribution of LLMs is therefore practical for analyzing and enhancing their veracity.

Existing studies typically approach the factuality detection task in the question-answering paradigm, which can be grouped into two families: *text-driven methods* and *KG-driven methods*, depending on the derivation of test cases. Text-driven methods utilize natural language text including news, summaries, or claims, to explore the absent knowledge of LLMs (Honovich et al. 2022; Durmus, He, and Diab 2020; Honovich et al. 2021). However, the high redundancy of text corpora could lead to wasted fact coverage during testing. Consider Figure 1, where *Text #1* explains a lot but elaborates on the sole fact that *LeBron James is a basketball player*. In contrast, KG-driven methods leverage knowledge graphs (KGs) to collect test cases by combining instance-level entities and relations (Pezeshkpour 2023; Kim et al. 2023; Agarwal et al. 2021; Wang, Wang, and Mao 2020; Pan et al. 2018). By utilizing concise triples, test cases of KG-driven methods shift to being more orthogonal and concentrated. For example in Figure 1, the simple triple (*LeBron James, Occupation, Basketball Player*) can replace the lengthy text representation.

In reality, **the above two groups of studies only conduct experiments using small-scale datasets in several typical domains**. Statistically, the coverage of the most extensive dataset is limited to around 30 commonsense topics, *e.g.*, movies and sports (Li et al. 2023). However, neglected fields, such as biology and plants, which are crucial for the artificial general intelligence (AGI) ability of LLMs, have not been adequately addressed (Xiong et al. 2023).

[*] These authors contributed equally.

[†] Corresponding authors.

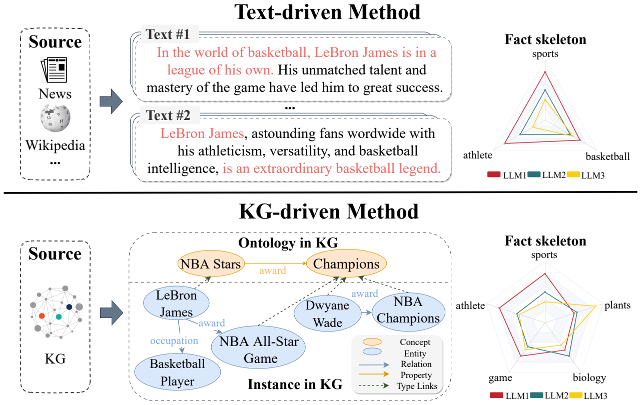


Figure 1: Comparison of text-driven and KG-driven methods $w.r.t.$ the test case of *Input: Is LeBron James a basketball player? Output: Yes*. The fact skeleton polygon depicts the level of LLMs' factuality towards different domains.

Moreover, previous research indicates that approximately 24% of fake information can be found on Facebook, while YouTube hosts around 27% (Beauvais 2022). Additionally, **the instance-level triples within the knowledge graph also contain a certain portion of un-facts**. For instance, a manual evaluation of the YAGO knowledge graph reveals that nearly 5% of the total triples are classified as inaccurate (Hoffart et al. 2013). Regarding these problematic datasets as golden benchmarks might further degrade the evaluation effect.

To address the above concerns, we argue that **the absent facts of LLMs can expose the nature of ontology consistency**. By examining the ontology-level structure (Studer, Benjamins, and Fensel 1998), we can identify triples that LLMs consider un-factual within KGs, revealing potential errors in LLMs' intrinsic hallucination. Typically, in Figure 1, the two un-factual triples *(LeBron James, award, NBA All-Star Game)* and *(Dwyane Wade, award, NBA Champions)* hold the same ontology structure *(NBA Stars, award, Champions)*. Furthermore, **utilizing ontology-level triples as probes can enhance the reliability of test cases**. This is because instance-level triples encapsulated within the same ontology tend to preserve a substantial proportion of true facts, despite the presence of occasional falsehoods. In this paper, we propose OntoFact, a robust ontology-driven factual detection framework, which shapes the factual detection task by employing discriminative questions ($i.e.$, with the answers `Yes` or `No`) to LLMs. Specifically, five representative KGs varying in English and Chinese are first adopted as benchmarks, involving 364-31,353 ontology-level and 36,250-707,758 instance-level triples. To adaptively produce error-prone test cases, a sophisticated ontology-driven reinforcement learning (ORL) mechanism is devised to interact between the ontology and corresponding instances. Such ORL mechanism guides the navigation of KGs and incrementally unveils factual errors, assembling un-factual ontologies into a fantastic skeleton for analyzing the unexpected hallucinations of LLMs. Furthermore, empirical efforts demonstrate LLMs are prone to reply `Yes` rather than `No`. Thus, we introduce a hallucination-free detection (HFD) strategy to tackle the overconfidence of LLMs and ensure fair comparisons between baselines.

We evaluate our OntoFact framework with 32 dominant LLMs, ranging from 1B to 175B parameters. Experimental results demonstrate the superior effectiveness of our method in detecting facts, predicting error rates, etc. Overall, existing LLMs lack factuality, with even the best-performing ChatGPT exhibiting a high factual error rate of 50.1% and 22.7% in the general and biomedical domains, respectively. Moreover, the ORL mechanism achieves F1 scores ranging from 70% to 90% in un-fact prediction with 35.29% to 63.12% time-reduction. To sum up, the contributions of this paper are four-fold:

- We argue that the ontology of KGs can reflect the fact skeleton of LLMs profoundly, and propose a robust ontology-driven factual detection framework OntoFact.
- We devise a novel reinforcement learning mechanism ORL, which can probe the error-prone ontologies and instances in KGs, and produce large-scale test cases.
- We conduct extensive experiments varying in LLMs and datasets, demonstrating our method's effectiveness in error prediction, fact detection, and learning efficacy.
- We open source 5 large-scale and wide-ranging fact-detection benchmarks to facilitate future research[1], and offer feasible insights to tackle LLMs' hallucination.

Method

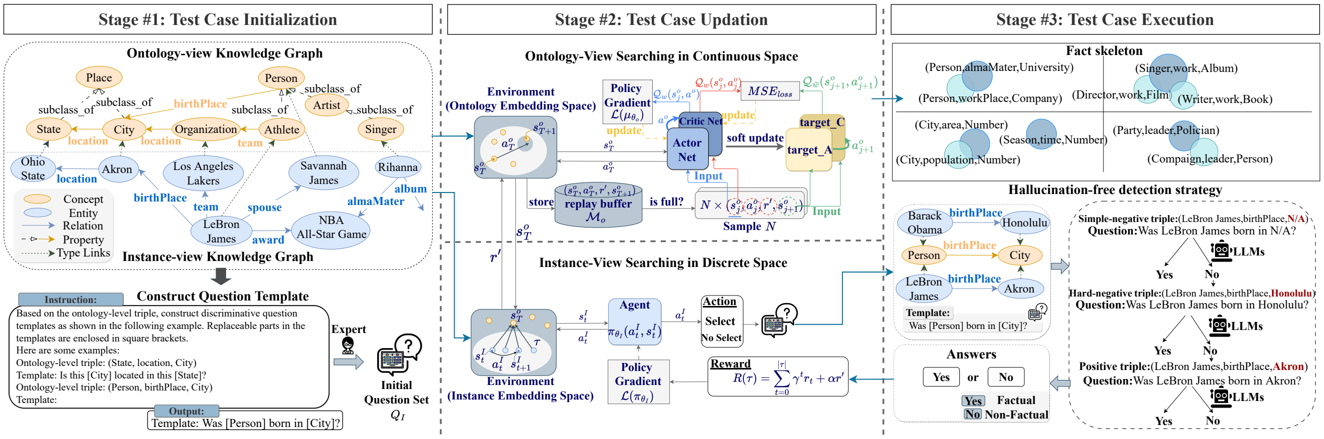
Suppose a specific KG $\mathcal{G} = \{\mathcal{G}_f, \mathcal{G}_o\}$ is given as the benchmark, where $\mathcal{G}_f = \{(h_f, r_f, t_f)\}$ and $\mathcal{G}_o = \{(h_o, r_o, t_o)\}$ stand for the instance and ontology sub-graph, while $h_\sim$, $r_\sim$, $t_\sim$ ($\sim \in \{f, o\}$) denote the head, relation, tail of a triple, respectively. The goal of factuality detection is to gather missing factual triples $\mathcal{T}_{mft} \subseteq \mathcal{G}_f$. Besides, the corresponding un-fact ontology skeleton $\mathcal{T}_{uos} \subseteq \mathcal{G}_o$ is also expected.

Figure 2 illustrates OntoFact's overall architecture. In the first stage, OntoFact initializes test cases by combining single instance-level triples. In the second stage, OntoFact leverages the ORL mechanism to wander along KG towards widely-range ontologies and instances, producing error-prone test cases adaptively. In the last stage, OntoFact feeds test cases into the hallucination-free detection module to obtain unbiased results.

Test Case Initialization

Following the previous study (Qin et al. 2023), the factuality assessment can be performed as discriminative question-answering (QA), in which the answer falls within $\{$`Yes`, `No`$\}$. Formally, a set of samples can be denoted as $S = \{(Q_1, R_1), (Q_2, R_2), \cdots, (Q_{|S|}, R_{|S|})\}$, where Q_i and R_i refer to the i^{th} natural language question and corresponding golden reference. The fact rate Υ can be calculated by first feeding $Q = \{Q_1, Q_2, \cdots, Q_{|S|}\}$ into the LLM to generate

[1] https://github.com/seukgcode/OntoFact


Stage #1: Test Case Initialization
Ontology-view Knowledge Graph
Place
Person
subclass_of
subclass_of
birthPlace
subclass_of
Artist
subclass_of
State
City
Organization
Athlete
Singer
location
location
team
Ohio State
Akron
location
Los Angeles Lakers
Savannah James
Rihanna
album
birthPlace
team
spouse
almaMater
LeBron James
award
NBA All-Star Game
Concept
Entity
Relation
Property
Type Links
Instance-view Knowledge Graph
Construct Question Template
Instruction:
Output:
Expert
Initial Question Set Q_I
Stage #2: Test Case Updation
Ontology-View Searching in Continuous Space
Environment (Ontology Embedding Space)
Policy Gradient $\mathcal{L}(\mu_{\theta_o})$
$\mathcal{Q}_w(s_j^o, a^o)$
$\mathcal{Q}_w(s_j^o, a_j^o)$
MSE_{loss}
$\mathcal{Q}_{\bar{w}}(s_{j+1}^o, a_{j+1}^o)$
a^o
Critic Net
update
update
Actor Net
soft update
target_C
target_A
a_{j+1}^o
s_{T+1}^o
a_T^o
s_T^o
s_T^o
a_T^o
Input
store
$(s_T^o, a_T^o, r', s_{T+1}^o)$
replay buffer $\mathcal{M}_o$
is full?
$N \times (s_j^o, a_j^o, r', s_{j+1}^o)$
Input
Sample N
s_T^o
r'
Instance-View Searching in Discrete Space
s_T^o
τ
s_t^I
a_t^I
s_{t+1}^I
Environment (Instance Embedding Space)
s_t^I
a_t^I
Agent
$\pi_{\theta_I}(a_t^I, s_t^I)$
a_t^I
Action
Select
No Select
Policy Gradient $\mathcal{L}(\pi_{\theta_I})$
Reward
$R(\tau) = \sum_{t=0}^{|\tau|} \gamma^t r_t + \alpha r'$
Stage #3: Test Case Execution
Fact skeleton
(Person,almaMater,University)
(Person,workPlace,Company)
(Singer,work,Album)
(Director,work,Film)
(Writer,work,Book)
(City,area,Number)
(Season,time,Number)
(City,population,Number)
(Party,leader,Polician)
(Compaign,leader,Person)
Hallucination-free detection strategy
Barack Obama
birthPlace
Honolulu
Person
birthPlace
City
LeBron James
birthPlace
Akron
Template:
Was [Person] born in [City]?
Answers
Yes
or
No
Yes Factual
No Non-Factual
Simple-negative triple:(LeBron James,birthPlace,N/A)
Question:Was LeBron James born in N/A?
LLMs
Yes
No
Hard-negative triple:(LeBron James,birthPlace,Honolulu)
Question:Was LeBron James born in Honolulu?
LLMs
Yes
No
Positive triple:(LeBron James,birthPlace,Akron)
Question:Was LeBron James born in Akron?
LLMs
Yes
No

Based on the ontology-level triple, construct discriminative question templates as shown in the following example. Replaceable parts in the templates are enclosed in square brackets.
Here are some examples:
Ontology-level triple: (State, location, City)
Template: Is this [City] located in this [State]?
Ontology-level triple: (Person, birthPlace, City)
Template:

Template: Was [Person] born in [City]?

Figure 2: Overview of OntoFact, where arrow between stages indicates the data flow from the previous stage to the next one.

answers $A = \{A_1, A_2, \cdots, A_{|S|}\}$, and then examining the consistency between A and R.

$$\Upsilon(A, R) = \frac{\sum_{i=1}^{|S|} \mathbb{1}[A_i \equiv R_i]}{|S|} \tag{1}$$

where $\mathbb{1}[\cdot]$ denotes the indicator operator.

In our work, the KGs are employed as the factual benchmarks to generate Q. Regarding the upper region of Stage #1 in Figure 2, we map each ontology-level triple with multiple corresponding instance-level triples, to construct the question templates and initialize the test cases. Based on this consideration, the ontology-level triples $\mathcal{T}_o = \{(C_h, r, C_t) | (h, r, t) \in \mathcal{G}_f \text{ and } h \in C_h \text{ and } t \in C_t\}$ are supplemented into the initial ontology sub-graph $\mathcal{G}_o$, where $C_h \subseteq \{h_o\}|_{o=1}^{|\mathcal{G}_o|}$, $C_t \subseteq \{t_o\}|_{o=1}^{|\mathcal{G}_o|}$, and (C_h, r, C_t) denotes the ontology-level triple corresponded with instance-level triple (h, r, t). For each (C_h, r, C_t) in $\mathcal{T}_o$, we first generate judgemental question templates with the help of ChatGPT as shown in the bottom region of Stage #1 in Figure 2. Considering an ontology-level triple $(Person, birthPlace, City)$, the question template could be *Was [Person] born in [City]?*. Then, filling up the above template with corresponding instance triples, *e.g.* *Was LeBron James born in Akron?*, yields initial question set Q_I.

Test Case Updation

The ORL mechanism (Algorithm 1) is fed into the initial Q_I, in order to produce large-scale test cases Q_E under ontology and instance views of KGs. The ontology-view agent guides the KGs' macroscopic walking direction, while the instance-view agent elicits error-proneness triples as the test case. Besides, the instance-view agent offers rewards for ontology-view agents to help identify un-factual ontologies efficiently.

Instance-View Searching in Discrete Space. Firstly, we sample ontology-level triples $\mathcal{T}_s = \{(C_{h_1}, r_1, C_{t_1}), \cdots, (C_{h_m}, r_m, C_{t_m})\}$ from $\mathcal{T}_o$. Then, for each ontology-level triple $(C_h, r, C_t) \in \mathcal{T}_s$, we randomly sample several corresponding instance-level triples $\mathcal{T}_{(C_h, r, C_t)} = \{(h, r, t) | (h \in C_h \text{ and } t \in C_t)\}$, and regard instance-level traversal process as a finite Markov decision process (MDP). At the t^{th} time-step, the state s_t^I is updated as embeddings concatenation *w.r.t.* current instance triple (h, r, t) (Lines 1-9 in Algorithm 1). Such embeddings of concepts $\mathbf{C}$, properties $\mathbf{R}^o$, entities $\mathbf{E}$, and relations $\mathbf{R}^I$ are learned by an off-the-shelf model JOIE (Hao et al. 2019). The instance-view agent delivers an action $a_t^I = \{0, 1\}$ to decide whether the triple (h, r, t) is chosen as a test case. The policy function can be formulated as follows:

$$\pi_{\theta_I}(a_t^I, s_t^I) = a_t^I \cdot f_{\theta_I}(s_t^I) + (1 - a_t^I) \cdot (1 - f_{\theta_I}(s_t^I)) \tag{2}$$

where $f_{\theta_I}(\cdot)$ implemented by a multi-layer perceptron (MLP) denotes the probability of (h, r, t) selected.

Moreover, two types of rewards, *i.e.*, immediate reward r_t and delayed reward r', are introduced (Lines 10-15 in Algorithm 1). The immediate reward activates when LLMs' answer is consistent with outputted action, while the delayed reward gives feedback after an MDP round. The instance-view agent aims to maximize the total reward of sampled triples $\mathcal{L}(\pi_\theta) = \mathbb{E}_{\tau \sim \mathcal{T}}[R(\tau)]$ (Lines 16-17 in Algorithm 1). Therefore, the instance-view agent can be updated by the policy gradient algorithm that can be formalized as follows:

$$\begin{aligned} \nabla \mathcal{L}(\pi_{\theta_I}) &= \mathop{\mathbb{E}}_{\tau \sim \mathcal{T}_s} \left[\sum_{t=1}^{|\tau|} \nabla_{\theta_I} \log \pi_{\theta_I}(a_t^I, s_t^I) R(\tau) \right] \\ &\approx \frac{1}{|\mathcal{T}_s|} \sum_{\tau \sim \mathcal{T}_s} \sum_{t=1}^{|\tau|} R(\tau) \nabla_{\theta_I} \log \pi_{\theta_I}(a_t^I, s_t^I) \end{aligned} \tag{3}$$

Ontology-View Searching in Continuous Space. Inspired by the DDPG algorithm (Lillicrap et al. 2015), we design an actor-critic network to increase the error-proneness on LLMs' traversal of ontology-level triples. Specifically, a replay buffer $\mathcal{M}_o$ of transitions $(s_T^o, a_T^o, r_T', s_{T+1}^o)$ is first collected, where the T^{th} time state s_t^o is also updated by embeddings concatenation of current ontology-level triples (C_h, r, C_t). Then, feeding state s_T^o into novel actor network $\mu_{\theta_o}(s_T^o)$ can produce corresponding actions a_T^o. Different from the discrete searching of instance view, ontology-view

Algorithm 1: Ontology-Driven Reinforcement Learning

```
Input: Initial questions set $Q_I$. The embeddings of entities $\mathbf{E}$, relations $\mathbf{R}^I$, concepts $\mathbf{C}$, and properties $\mathbf{R}^o$. LLM, policy network $f_{\theta_I}$, parameters $\gamma, \alpha, \beta$, threshold $c$, actor-critic network $\mu_{\theta_o}$, $\mathcal{Q}_w$, $\mu_{\bar{\theta}}$, $\mathcal{Q}_{\bar{w}}$, and sampled ontology-level triples $\mathcal{T}_s$.
Output: Policy network $\pi_{\theta_I}$, actor-critic network $\mu_{\theta_o}$, $\mathcal{Q}_w$, $\mu_{\bar{\theta}}$, $\mathcal{Q}_{\bar{w}}$.
1: $Visit = \varnothing$
2: $\tau \leftarrow$ Random select ontology-level triple $(C_h, r, C_t)$ from $\mathcal{T}_s$.
3: $Visit[0] \leftarrow \tau$.
4: Initialize the replay buffer $\mathcal{M}_o = \varnothing$.
5: for $i \leftarrow 0$ to $|\mathcal{T}_s| - 1$ do
6:     $a^I[0:|\tau|], A[0:|\tau|], r[0:|\tau|] \leftarrow 0$
7:     for $j \leftarrow 0$ to $|\tau| - 1$ do
8:         $h_f, r_f, t_f \leftarrow \tau[j]$.
9:         $s^I_j \leftarrow [\mathbf{E}_{h_f}; \mathbf{R}^I_{r_f}; \mathbf{E}_{t_f}]$
10:        Obtain current action $a^I[j]$ using $s^I_j$ based on Equ 2.
11:        $A[j] \leftarrow \mathrm{LLM}(Q_I^{(h_f, r_f, t_f)})$
12:        $r[j] \leftarrow 2 \times (a^I[j] \oplus A[j]) - 1$
13:    end for
14:    $err_L \leftarrow \frac{cnt(A[A=0])}{|\tau|}$,   $err_A \leftarrow \frac{cnt(a^I[a^I=1])}{|\tau|}$
15:    $r' = (err_A - c)\mathbf{sgn}(err_L - c)$
16:    $R(\tau) = \sum_{t=0}^{|\tau|-1} \gamma^t r[t] + \alpha r'$
17:    Update $\pi_{\theta_I}(\cdot)$ based on Equ. 3.
18:    $s^o_i \leftarrow [\mathbf{C}_\tau; \mathbf{R}^o_\tau; \mathbf{C}_\tau]$
19:    $a^o_i \leftarrow \mu_{\theta_o}(s^o_i)$
20:    $\tau, s^o_{i+1} \leftarrow \mathrm{argmin}_{k \in (\mathcal{T}_s - Visit)} ||a^o_i - \mathbf{k}||_2$
21:    $\mathcal{M}_o \leftarrow \mathcal{M}_o \cup \{(s^o_i, a^o_i, r', s^o_{i+1})\}$
22:    if $\mathcal{M}_o$ is full then
23:        Update $\mathcal{Q}_w(\cdot)$ and $\mu_{\theta_o}(\cdot)$ based on Equ. 4 and Equ. 6.
24:        $\bar{\theta} \leftarrow \beta\theta_o + (1-\beta)\bar{\theta}$,   $\bar{w} \leftarrow \beta w + (1-\beta)\bar{w}$
25:    end if
26:    $Visit[i] \leftarrow \tau$
27: end for
```

RL conducts action in a continuous action space. Since the embedding of generated action a^o_T might not be absolutely equal to existing ontology-level triples, we select the closest one as the next state s^o_{T+1}. The above process is detailed in Lines 18-21 of Algorithm 1. To evaluate the effectiveness of action a^o_T, we leverage the critic network $\mathcal{Q}_w(s^o_T, a^o_T)$ to output a score $q \in [0, 1]$, predicting the LLMs' un-factuality proportion on the current ontology-level triple. Such actor and critic networks are also implemented by the MLP.

Moreover, to stabilize the training process, we design the target actor and critic network with parameters $\bar{\theta}$ and $\bar{w}$, which holds the same architecture as the original actor and critic network. To optimize the critic network $\mathcal{Q}_w(\cdot)$, we first randomly sample M transitions from $\mathcal{M}_o$. Then, the mean squared error (MSE) loss is minimized, which can be formulated as follows:

$$\mathcal{L}(\mathcal{Q}_w) = \frac{1}{|M|} \sum_i \left(y_i - \mathcal{Q}_w\left(s^o_i, a^o_i\right)\right)^2 \tag{4}$$

$$y_i = r'_i + \gamma \mathcal{Q}_{\bar{w}}\left(s^o_{i+1}, \mu_{\bar{\theta}}\left(s^o_{i+1}\right)\right) \tag{5}$$

where $\gamma \in [0, 1]$ is a discount. For the actor network, the output score of $\mathcal{Q}_w(\cdot)$ can be regarded as a reward, which can be maximized to optimize parameters θ_o as follows:

$$\mathcal{L}(\mu_{\theta_o}) = \max_{\theta_o} \mathop{\mathbb{E}}_{s^o_T \sim M}\left[\mathcal{Q}_w\left(s^o_T, \mu_{\theta_o}\left(s^o_T\right)\right)\right] \tag{6}$$

For the target actor and critic network, we use *soft* parameter optimization strategy (Line 24 in Algorithm 1). Finally, incorporating ontology-view and instance-view agents, ORL can output potentially un-factual test cases Q_E. Specifically, for instance-level triples (h, r, t) under corresponding ontology-level triple $(C_h, r, C_t) \in \mathcal{T}_o - \mathcal{T}_s$, those with the selection probability higher than 0.5 are included, *i.e.*, $Q_E = \{Q_I^{(h,r,t)} | f_{\theta_I}(s^I_t(h, r, t)) \geq 0.5\}$ can be obtained.

Test Case Execution

To alleviate the overconfidence of LLMs, we devise a hallucination-free detection (HFD) strategy to execute test cases Q_E and boost the robustness of LLMs' `yes` or `no` response. Especially, take ontology-level triple (*Person*, *birthPlace*, *City*) and its instance-level triple (*LeBron James*, *birthPlace*, *Akron*), (*Barack*, *birthPlace*, *Honolulu*) as an example (Figure 2 Stage #3), we first construct the simple negative sample by replacing the tail entity *Akron* with a meaningless character `N/A`. If LLMs respond `yes`, it is clear that they lack that factual information. If not, we proceed to construct the hard negative sample by replacing tail entity with *Honolulu* under the same ontology. If answer is still `No`, the original positive triple (*LeBron James*, *birthPlace*, *Akron*) is adopted. Overall, the output depends on three aspects:

$$\mathrm{LLM}(Q_E) = \mathbb{1}[s_1 \equiv \texttt{No}] \wedge \mathbb{1}[s_2 \equiv \texttt{No}] \wedge \mathbb{1}[s_3 \equiv \texttt{Yes}] \tag{7}$$

where $\mathbb{1}[\cdot]$ denotes the indicator operator, $\wedge$ imply the logical conjunction operator, and s_1, s_2, and s_3 denote LLMs' answer $w.r.t.$ simple negative, hard negative, and positive samples, respectively. During the test case execution, LLMs may generate responses that are not simply `Yes` or `No`. In response to this situation, a natural language inference (NLI) model is introduced for auxiliary judgment, converting `Yes` or `No` into `entailment` or `contradiction`.

Finally, the missing factual triples set $\mathcal{T}_{mft}$ can be established by triples with a finite answer `No`. Meanwhile, for each ontology-level triple, we calculate its error rate by the corresponding instance triple in $\mathcal{T}_{mft}$ and then add those ontology-level triples with an error rate greater than 50% to the un-fact ontology skeleton set $\mathcal{T}_{uos}$.

Experiments

Experimental Setup

In this section, we describe datasets and benchmarks, typical LLMs, evaluation metrics, and implementation details.

Datasets and Benchmarks. To investigate the LLMs' factuality for general knowledge, we employ three large-scale KGs, where DBpedia (Lehmann et al. 2015) and YAGO 4.5 (Pellissier Tanon, Weikum, and Suchanek 2020) are in English (ENG) while CN-DBpedia (Xu et al. 2017) is in Chinese (CNS). Regarding specific domains, we adopt a bilingual biomedical KG (Yu et al. 2022), *i.e.*, BIOS 2.2 (ENG) and BIOS 2.2 (CHS). Table 1 provides the statistics.

LLMs. For English baselines, we choose 20 LLMs (1B to 175B parameters) from 7 unique LLMs-families, including ChatGPT-175B (Ouyang et al. 2022), LLaMA (Touvron et al. 2023), T0pp (Sanh et al. 2022), OPT (Zhang

Dataset	#Ontologies	#Instances
	Trip./Conc./Prop.	Trip./Ent./Rel.
DBpedia	31,353/405/459	636,532/581,374/459
YAGO	15,876/1,987/51	707,758/622,651/51
CN-DBpedia	16,963/469/657	634,450/725,714/657
BIOS 2.2 (ENG)	430/27/12	42,858/40,561/12
BIOS 2.2 (CHS)	364/27/12	36,250/33,729/12

Table 1: Dataset statistics (Trip, Conc, Prop, Ent, Rel denote triples, concepts, properties, entities, and relations, respectively).

et al. 2022), BLOOM (Scao et al. 2022), GPT (Radford et al. 2019), FLAN-T5 (Chung et al. 2022). For Chinese baselines, 12 LLMs (6B to 175B parameters) are selected, including ChatGPT-175B (Ouyang et al. 2022), ChatGLM (Zeng et al. 2023), and LLaMA (Touvron et al. 2023).

Evaluation Metrics. To metric the factuality of LLMs, we leverage the exhausting strategy combined with the HFD strategy in all benchmarks to calculate the error proportion (EP) of ontology-level triples. For each ontology-level triple, if negative answers of correlated instance-level triples exceed 50%, an un-fact triple occurred. For ORL, which is designed to predict whether a given ontology-level triple is an un-factual domain for LLMs, it can be viewed as a binary classification task. Therefore, we measure the performance of ORL utilizing commonly used accuracy ACC, precision P, recall R, and the corresponding F1-score. Specifically, we randomly select one-third of datasets for training, and calculate the above metric of ORL on the rest two-thirds parts.

Implementation Details. All experiments are implemented on the NVIDIA A100 (80GB) GPU. In all experiments, for the embedding of instance graphs, the embedding size of entities and relations is 300 and 100, respectively. For the embedding of ontology graphs supplemented with ontology-level triples, the embedding size of concepts and properties is 100. Besides, the instance-view agent is the two-layer MLP in which the activation function of the hidden layer is ReLU, and the number of neural units is kept consistent with the size of the input dimensions. The value of γ in the total reward $R(\tau)$ for each ontology-level triple is 0.95, and the value of α is 12.0. For the threshold c, it is set to 0.5. In the ontology-view agent, both the actor network and the critic network are the two-layer MLP in which the activation function of the hidden layer is ReLU, and the number of neurons in the hidden layer is kept consistent with the size of the input dimension. The size of M in the optimized ontology-view agent is 2. The value of γ used in the optimized criticism network is 0.95. The value of β used in the soft optimization of the target action-critic network is 0.001. Moreover, three Adam optimizers with a learning rate of $1e-4$ are used in ORL to optimize the actor network, the critic network in the ontology-view agent, and the instance-view agent, respectively. In the test case execution, for LLMs in the English datasets and Chinese datasets, we utilize the t5_xxl_true_nli_mixture[2] and the Erlangshen-MegatronBert-1.3B-NLI[3], respectively.

Main Result

The main results are reported in Table 2 and Table 3, and we have the following observations and conclusions.

First, the factuality of LLMs is generally poor, even lower than random prediction. Specifically, factual error rates of LLMs on general domain datasets DBpedia, YAGO, and CN-DBpedia are 51.6%-100%, 53.8%-100%, and 33.9%-100%, respectively. In statistics, ChatGPT and BLOOM$_{\text{3B}}$ hold the best and worst with average EP scores of 50.1% and 100.0%, respectively. For the domain-specific dataset BIOS, ChatGPT achieves a lower error rate with 9.1% in BIOS (CHS) and 36.3% BIOS (ENG), while BLOOM$_{\text{3B}}$ has a high error rate (100%) on BIOS (ENG). In comparison with raw LLaMA$_{\text{7B}}$, BenTsao$_{\text{7B}}$ enhances a considerable factuality (16.4% decline in EP), which could benefit from fine-tuning with medical datasets. These results demonstrate the different awareness of specific knowledge between different LLMs, which can provide support for personalized knowledge injection of LLMs.

Second, LLMs with larger parameters tend to offer higher factuality. Regarding the three English datasets, ChatGPT with 175B parameters yields an average error rate of 50.9%, while the result of the second largest model LLaMA$_{\text{13B}}$ is 73.1%, which exists a gap of more than 22.2%. For more details, we investigate the LLMs' factuality in different intervals in Figure 3, where larger LLMs tend to lower error intervals (50%-69%). Moreover, the performance of the same LLMs-family under small-to-large parameter scales further echoes the above viewpoint. Incrementally, OPT$_{\text{6.7B}}$ improves the average factual accuracy by 2.1% and 3.9% over its 2.7B and 1.3B versions. However, OPT$_{\text{13B}}$ has a 14.6% increase in average un-factual rate compared to its 6.7B version due to significant overconfidence. Another notable observation is that the EP of BLOOM$_{\text{7B}}$ and LLaMA$_{\text{13B}}$ outperform ChatGPT on YAGO dataset by 10.9% and 10.4%, respectively, despite parameters being 10-100 times smaller. This is primarily because ChatGPT refuses to respond to numerous of the historical celebrity details included in YAGO.

Third, the ORL mechanism demonstrates promising error prediction and fact detection ability. On DBpedia, LLaMA$_{\text{7B}}$ has a high error rate of 99.7%, while ORL achieves a high prediction accuracy of 99.8% and F1 score of 99.9%. Meanwhile, LLaMA$_{\text{13B}}$ obtains a lower error rate (74.6%), and the corresponding ORL's prediction accuracy also drops to 76.5%. This demonstrates a significant positive correlation between ORL's error prediction performance and LLMs' inherent error rate. Besides, ORL achieves a recall rate as high as 50.0%-100% and F1 score as high as 56.9%-100% in domains that are not seen during training. Such ability of ORL to explore unknown domains can be generalized to analyze the unexpected hallucination of LLMs.

Last, ChatGPT is surprisingly more factual in Chinese benchmarks than in English ones. Specifically, the average EP of ChatGPT on the English general dataset is 50.9%,

[2] https://huggingface.co/google/t5_xxl_true_nli_mixture

[3] https://huggingface.co/IDEA-CCNL/Erlangshen-MegatronBert-1.3B-NLI

LLMs	DBpedia					YAGO					BIOS 2.2 (ENG)				
	EP↓	ACC↑	P↑	R↑	F1↑	EP↓	ACC↑	P↑	R↑	F1↑	EP↓	ACC↑	P↑	R↑	F1↑
Random	54.3	-	-	-	-	54.3	-	-	-	-	54.0	-	-	-	-
$\text{LLaMA}_{7\text{B}}$	99.7	99.8	99.8	99.9	99.9	95.5	94.0	94.2	99.7	96.9	97.4	97.8	97.8	100	98.9
$\text{LLaMA}_{13\text{B}}$	<u>74.6</u>	76.5	80.2	90.3	85.0	<u>54.3</u>	82.6	87.9	78.9	83.2	90.5	87.4	87.4	100	93.3
$\text{Vicuna}_{13\text{B}}$	85.6	88.9	90.5	95.3	92.8	80.3	90.3	92.3	96.5	94.4	92.7	96.6	96.5	97.8	97.1
$\text{Alpaca}_{7\text{B}}$	79.3	86.6	88.1	90.7	89.4	72.8	86.5	89.3	92.5	90.9	95.1	94.8	94.8	100	97.3
$\text{T0pp}_{11\text{B}}$	93.1	93.9	94.6	99.1	96.8	69.2	82.2	89.8	74.9	81.7	92.8	90.0	90.0	100	94.7
$\text{OPT}_{1.3\text{B}}$	85.9	88.7	90.2	95.8	92.9	79.7	90.7	93.3	96.1	94.7	100	100	100	100	100
$\text{OPT}_{2.7\text{B}}$	84.2	86.3	88.3	96.1	92.0	78.6	83.6	87.3	90.5	88.9	97.2	96.1	96.1	100	98.0
$\text{OPT}_{6.7\text{B}}$	80.6	87.8	89.3	93.2	91.2	74.1	88.7	91.5	93.8	92.6	99.1	98.7	98.7	100	99.3
$\text{OPT}_{13\text{B}}$	98.3	96.8	99.4	95.8	97.6	99.2	99.4	99.4	100	99.7	100	100	100	100	100
$\text{BLOOM}_{1.1\text{B}}$	99.9	99.9	99.9	100	99.9	99.9	99.9	99.9	100	99.9	91.4	84.8	85.5	99.0	91.8
$\text{BLOOM}_{1.7\text{B}}$	98.1	98.1	98.2	99.9	99.1	92.5	93.7	95.1	97.9	96.5	86.7	75.7	87.4	81.7	84.5
$\text{BLOOM}_{3\text{B}}$	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
$\text{BLOOM}_{7\text{B}}$	76.8	83.1	87.7	90.3	89.0	**53.8**	84.9	85.7	79.7	82.6	82.6	96.1	96.1	100	98.0
$\text{GPT-Neo}_{1.3\text{B}}$	93.4	94.2	94.4	99.7	97.0	92.7	93.8	93.6	94.1	93.8	100	100	100	100	100
$\text{GPT2-XL}_{1.5\text{B}}$	92.5	93.6	95.2	98.7	96.9	90.3	91.2	94.7	95.0	94.8	84.7	80.4	81.7	97.3	88.8
$\text{GPT-Neo}_{2.7\text{B}}$	94.8	94.4	95.3	99.0	97.1	87.4	91.4	93.1	97.7	95.3	77.2	81.7	82.1	97.1	89.0
$\text{GPTJ}_{6\text{B}}$	96.8	98.2	99.1	97.5	98.3	93.4	91.6	91.9	95.8	93.8	98.8	95.7	99.5	96.0	97.7
$\text{FLAN-T5-XL}_{3\text{B}}$	96.8	97.2	98.8	99.6	99.2	88.9	87.5	90.1	91.4	90.7	91.6	86.5	86.5	100	92.8
$\text{FLAN-T5-XXL}_{11\text{B}}$	94.2	93.9	94.3	99.5	96.8	80.0	92.4	95.1	95.2	95.1	<u>67.8</u>	63.0	63.0	98.5	76.9
$\text{ChatGPT}_{175\text{B}}$	**51.6**	73.2	80.3	64.2	71.4	64.7	95.2	95.3	98.2	96.7	**36.3**	72.9	66.1	50.0	56.9

Table 2: Comparison of English LLMs, where ↑ indicates a larger value is preferred and ↓ means the opposite. `Random` refers to randomly classifying ontology-level triples into un-fact. EP(%) denotes the error proportion of ontology-level triples, where bold indicates the best performance and underline denotes the second-best performance. ACC(%) represents the ORL's accuracy in predicting ontology-level triples. P(%), R(%), and F1(%) denote the ORL's precision, recall, and F1 score, respectively.

LLMs	CN-DBpedia					BIOS 2.2 (CHS)				
	EP↓	ACC↑	P↑	R↑	F1↑	EP↓	ACC↑	P↑	R↑	F1↑
Random	54.3	-	-	-	-	54.4	-	-	-	-
$\text{ChatGLM}_{6\text{B}}$	53.4	86.4	86.0	84.5	85.3	83.5	79.0	79.0	100	88.3
$\text{ChatGLM2}_{6\text{B}}$	51.6	84.3	86.4	83.8	85.1	<u>48.6</u>	50.8	48.5	94.3	64.1
$\text{CHSLAAp}_{7\text{B}}$	88.7	87.8	87.8	99.9	93.5	81.9	80.3	84.7	100	91.7
$\text{BELLE}_{7\text{B}}$	100	100	100	100	100	100	100	100	100	100
$\text{BELLE}_{13\text{B}}$	86.7	92.2	93.3	97.6	95.4	52.5	80.5	97.6	64.4	77.6
$\text{Baichuan}_{7\text{B}}$	99.0	98.3	98.8	99.5	99.1	89.0	88.7	89.0	99.7	94.0
$\text{Baichuan}_{13\text{B}}$	<u>49.8</u>	85.1	87.2	81.0	84.0	56.6	91.3	91.3	100	95.4
$\text{DoctorGLM}_{6\text{B}}$	93.6	95.8	96.7	98.2	97.4	100	100	100	100	100
$\text{BenTsao}_{7\text{B}}$	95.2	96.7	96.9	98.1	97.5	83.6	85.9	87.1	88.4	87.7
$\text{HuatuoGPT}_{7\text{B}}$	97.2	98.0	98.6	99.5	99.0	49.2	79.4	84.2	71.5	77.3
$\text{LLaMA}_{7\text{B}}$	99.7	99.1	99.8	99.3	99.5	100	100	100	100	100
$\text{LLaMA}_{13\text{B}}$	100	100	100	100	100	100	100	100	100	100
$\text{ChatGPT}_{175\text{B}}$	**33.9**	81.3	81.4	79.6	80.5	**9.1**	92.0	75.6	72.8	74.2

Table 3: Comparison of Chinese LLMs.

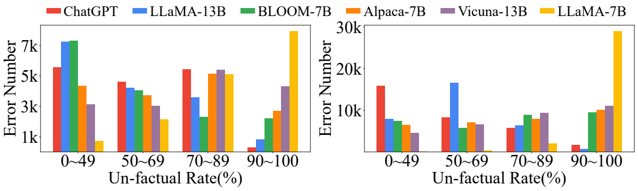


Figure 3: Statistics of the LLMs' un-facts on YAGO (left) and DBpedia (right) datasets.

which is 29.4% higher than that of the Chinese general dataset. On one hand, compared with Chinese, the wider application of English makes its knowledge broader. On the other hand, different language proportions used in the pre-training of LLMs could also differ in the support of Chinese and English. Typically, since LLaMA's training corpus includes limited Chinese corpora, its EP in Chinese datasets is significantly higher than in English datasets. However, for Chinese-LLaMA-Alpaca-$\text{pro}_{7\text{B}}$ ($\text{CHSLAAp}_{7\text{B}}$) using $\text{LLaMA}_{7\text{B}}$ as the base model and then continuing to pre-train it with large-scale Chinese language corpora, as well as $\text{Baichuan}_{7\text{B}}$, which holds the same model structure as $\text{LLaMA}_{7\text{B}}$ but is trained from scratch with large Chinese corpora, the EP decrease by 11.0% and 0.7% in CN-DBpedia, respectively, compared to raw $\text{LLaMA}_{7\text{B}}$.

Analysis Experiment

In this section, we conduct experiments to analyze ORL learning efficiency and LLM factual skeleton.

About ORL Learning Efficiency. To verify the efficiency of the ORL mechanism, we remove ORL and adopt the exhausting strategy (referred to w/o) to detect LLMs' un-factual ontology-level triples on DBpedia and YAGO. Particularly, one-third of the dataset is first fed to ORL for training, and the rest is used for comparison. It is worth noting that in the ORL time calculation, the training time of ORL is also included for fairness of comparison. Combining the

LLMs	DBpedia			YAGO		
	ORL(h)	w/o(h)	↓(%)	ORL(h)	w/o(h)	↓(%)
LLaMa$_{13B}$	32.51	60.73	46.47	21.97	44.09	50.17
Vicuna$_{13B}$	32.48	57.20	43.21	18.31	44.10	58.48
Alpaca$_{7B}$	13.27	24.95	46.82	18.71	50.73	63.12
T0pp$_{11B}$	8.14	16.68	51.20	13.37	20.66	35.29
BLOOM$_{7B}$	25.89	68.22	62.05	23.21	50.72	54.24
GPTJ$_{6B}$	17.54	35.06	49.97	21.64	43.49	50.24
ChatGPT$_{175B}$	109.53	267.04	58.98	209.96	330.75	36.52

Table 4: The graph search time (hour) of ORL, where ↓(%) indicates the time-reducing proportion with ORL.

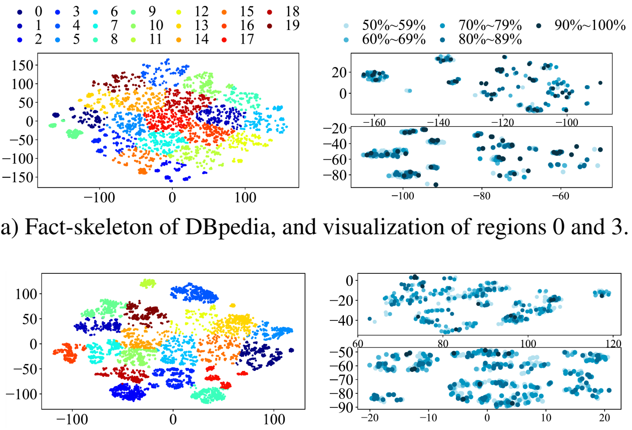
consuming time from Table 4 and the recall rate from Table 2, the ORL mechanism is able to reduce the evaluation budget by 35.29%-63.12%, while identifying 64.2%-99.1% of un-factual triples. These results simultaneously provide support for both the learning effect and efficiency of ORL.

About LLMs Factuality Skeleton. As shown in Figure 4, the principle of un-factual fact-skeleton is fantastic. On the one hand, the non-facts of LLMs show obvious aggregation and clustering (see the left part of Figure 4). On the other hand, the centers of each cluster are the highest un-factual ontology-level triples and gradually disperse to the lowest un-factual ones (see the right part of Figure 4). To further understand the generation process of such fact-skeleton, we further illustrate the Euclidean distance between two neighbor ontology-level triples during graph searching of ORL on YAGO in Figure 5. In the process of exploring un-factual domains, the ORL exhibits an interesting behavior characterized by periodic jumps and smooth periods of exploration. This behavior can be observed by examining the Euclidean distances between points in the ORL trajectory. On the whole, ORL first greedily touches the neighboring triples of afore-mined un-fact ones for a period, and then jumps to a remote location, expecting to explore more diverse un-factual domains. The fantastic skeleton is formed in the repeated interaction of ORL.

Case Study

In this section, we provide two case studies to help further understand our OntoFact.

Example of Hallucination-Free Detection. To demonstrate the effectiveness of the hallucination-free detection (HFD) strategy, we present relevant cases in Table 5. First, straightforward `prompt` containing implicit cues could lead to biased responses from LLMs. Regarding the first line, the phrases `with yes or no` and `with no or yes` guide LLMs to generate opposite answers for the same question. To tackle this commonly occurred issue in low- and mid-parameter (1B-13B) LLMs, we employ question prompts devoid of implicit answer cues. Secondly, the `Tense` of questions influences the factuality evaluation of LLMs. Regarding the second line, to address the tendency of ChatGPT to reject answering `present-tense` questions, we predominantly structure the question types in the `past tense`. Finally, the proposed `HFD` mechanism effectively mitigates overconfidence in LLMs. Regarding the third line, when directly asking OPT-13B judgment questions, it consistently responds with `yes` rather `no` than in most cases. By incorporating two types of negative samples (*e.g.*, `N/A` and `California`) for each positive sample, we can effectively prevent such biased factuality evaluation issues.



(a) Fact-skeleton of DBpedia, and visualization of regions 0 and 3.

(b) Fact-skeleton of YAGO, and visualization of regions 8 and 17.

Figure 4: Visualization of fact-skeleton for ChatGPT. Embeddings of ontology-level triples are dimension-reduced and clustered into 20 regions (left). Two random regions are detailed (right), where the color gradient from dark to light indicates the extent of LLMs' un-facts from high to low.

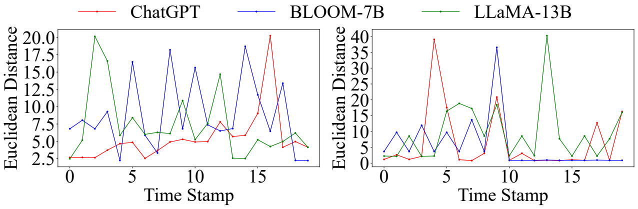


Figure 5: Visualization of the neighbors' Euclidean distance under ORL search on YAGO (left) and DBpedia (right).

Type	Question	Answer
SP	Please answer questions with **yes or no**: Was LeBron James born in Akron?	Yes
	Please answer questions with **no or yes**: Was LeBron James born in Akron?	No
T	**Did** LeBron James win NBA All-Star Game?	Yes
	Does LeBron James win NBA All-Star Game?	GU
	Was Alex Neri a member of Planet Funk?	Yes
	Is Alex Neri a member of Planet Funk?	GU
HFD	Was Allegheny Forest located in **N/A**?	Yes
	Was Allegheny Forest located in **California**?	Yes
	Was Allegheny Forest located in **Pennsylvania**?	Yes

Table 5: Examples of hallucination-free detection (HFD), where SP means straightforward prompt, T means tense of questions, and GU implies that LLMs give up answering.

Example of ORL Graph Search. To illustrate the graph search process of ORL between the ontology and instance level KGs, we conduct experiments using ChatGPT and

TS	**Ontology-Level Triples**	**ED**	**UR**
T_i	(**Plant**, **hybrid**, **Species**)	25.8	72.0
TS	**Instance-Level Triples**	**Action**	**Answer**
T_i^0	(Pixie mandarin, hybrid, Cam sành)	No Select	Yes
T_i^1	(Kalette, hybrid, Kale)	Select	No
⋮	⋮	⋮	⋮
T_i^{49}	(Haruka citrus, hybrid, Hyuganatsu)	*Select*	*Yes*
TS	**Ontology-Level Triples**	**ED**	**UR**
T_{i+1}	(**Plant**, **wineRegion**, **Place**)	8.8	76.7
TS	**Instance-Level Triples**	**Action**	**Answer**
T_{i+1}^0	(Assyrtiko, wineRegion, Greece)	Select	No
T_{i+1}^1	(Baco noir, wineRegion, Wisconsin)	Select	No
⋮	⋮	⋮	⋮
T_{i+1}^{49}	(Aleatico, wineRegion, Tuscany)	No Select	Yes

Table 6: Example of ORL graph search for ChatGPT on DBpedia at two time stamps, with TS as time stamp, ED as Euclidean distance, and UR as the un-fact rate of instance-level triples associated with the current ontology-level triple.

present the record in Table 6. At a specific time step, denoted as T_i, the ontology-view agent traverses a distance of 25.8, jumping from a remote domain to the plant-related field. The instance-view agent then identifies the associated instance-level triples, addressing the error-prone triples properly. Despite occasional mistakes in triple judgment ($e.g.$, at time T_i^{49}), the overall predictions remain within an acceptable range. At time stamp T_{i+1}, due to the clustering nature of factual errors, the ontology-view agent greedily searches for another neighbor ontology-level triple related to plants, located nearby with a distance of only 8.8. In conclusion, the instance-view agent systematically identifies the instance-level triples, one by one, in order to infer the factual probabilities of the corresponding ontology-level triples.

Discussion

KG Search Motivation KG Searching aims to efficiently mine ontology-level triples that are error-prone for LLMs based on knowledge graph embeddings (Hao et al. 2019; Liu et al. 2023; Shang et al. 2023), where the ontology-view agent is in the continuous space and the instance-view agent is in the discrete space. The motivation for setting continuous and discrete spaces depends on the number of search states. Typically, the general domain datasets constructed in this paper contain 15K-31K ontology-level triples, with an average of only 30-100 corresponding instance-level triples per ontology triple. Therefore, large ontology and small instance sub-graphs indicate the suitability of continuous and discrete space searching, respectively.

Explanation of HFD Hallucination-free Detection (HFD) alleviates over-confidence of LLMs through: (1) modifying the head or tail entity with N/A or other ontology-specific entities to produce negative samples, and (2) feeding additional negative samples into LLMs to obtain a robust response. Over 32 LLMs on 5 benchmarks, for each instance-level triple, modifying the head or tail entities of the same triple (two negative samples) yields an average error proportion (EP) improvement of 13% against only modifying a single entity (one negative sample).

Human Evaluation OntoFact adopts the discriminative Q&A (`Yes/No`) with a unique answer to model factuality assessment. To some extent, the setting of discriminative Q&A alleviates the issue of unfair evaluation performance measurement when the same entity has multiple representations, which commonly occurs with the commonly used evaluation metric, $e.g.$, Exact Match. However, the OntoFact proposed in this paper requires the NLI model for assisted judgment. Therefore, we further manually evaluate judgments made by the NLI model. Specifically, for all test cases, we sampled 500 test cases and compared the judgments of the NLI model with the manual evaluation of the generated response of LLMs. Results demonstrate a high agreement (97%) between machine (using the off-the-shelf NLI model) and human evaluation.

Related Work

The hallucinations of LLMs have gained significant attention due to their noxious impact on practical applications. According to Ji et al. (2023), hallucinations can be categorized as intrinsic or extrinsic (Maynez et al. 2020; Huang et al. 2021). Intrinsic hallucinations refer to conflicts between LLMs' output and the source, while extrinsic hallucinations involve unverifiable output. Previous works have analyzed the cause and interpretability of hallucinations in various aspects (McKenna et al. 2023; Zhang et al. 2023; Zheng, Huang, and Chang 2023). Another research line aims to evaluate LLMs' hallucinations in different NLP tasks, primarily focusing on assessing intrinsic hallucinations through factuality detection (Xie et al. 2021; Honovich et al. 2022; Mountantonakis and Tzitzikas 2023; Min et al. 2023; Pezeshkpour 2023). Generally, the factuality detection can be categorized into text-driven and KG-driven methods. On one hand, Honovich et al. (2022) and Xie et al. (2021) measure LLMs' factuality using standardized text corpora from diverse tasks. On the other hand, KG-driven methods generate test cases with concise entity-relation triples (Mountantonakis and Tzitzikas 2023). Min et al. (2023) introduce a biography KG to evaluate the factuality of LLMs. Pezeshkpour et al. (2023) employ information theory-based measurements to estimate the factual knowledge of LLMs. In summary, existing techniques fail to uncover the missing fact skeleton and provide a beneficial fact injection strategy for alleviating the hallucination, which is the core focus of our proposed work.

Conclusion

In this paper, we address the main challenges of the factuality evaluation of LLMs and present OntoFact, an adaptive un-facts detection framework with a sophisticated ORL mechanism, aiming to mine the ontology-level knowledge-lacking skeleton. Experimental results on a wide range of LLMs and datasets demonstrate the robustness and generalization of OntoFact in predicting errors, detecting facts, $etc.$ Moreover, we release five large-scale benchmarks that can be beneficial for relevant research.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by National Science Foundation of China (Grant Nos.62376057) and the Start-up Research Fund of Southeast University (RF1028623234). All opinions are of the authors and do not reflect the view of sponsors.

References

Agarwal, O.; Ge, H.; Shakeri, S.; and Al-Rfou, R. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In *ACL*.

Beauvais, C. 2022. Fake news: Why do we believe it? *Joint bone spine*, 89(4): 105371.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Chung, H. W.; Hou, L.; Longpre, S.; et al. 2022. Scaling instruction-finetuned language models. In *ICLR*.

Durmus, E.; He, H.; and Diab, M. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *ACL*.

Hao, J.; Chen, M.; Yu, W.; et al. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *SIGKDD*.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; and Weikum, G. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194: 28–61.

Honovich, O.; Aharoni, R.; Herzig, J.; et al. 2022. TRUE: Re-evaluating factual consistency evaluation. In *ACL*.

Honovich, O.; Choshen, L.; Aharoni, R.; et al. 2021. Q^2: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering. In *ACL*.

Huang, Y.; Feng, X.; Feng, X.; and Qin, B. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Kang, D.; and Hashimoto, T. 2020. Improved natural language generation via loss truncation. In *ACL*.

Kim, J.; Park, S.; Kwon, Y.; Jo, Y.; Thorne, J.; and Choi, E. 2023. FactKG: Fact Verification via Reasoning on Knowledge Graphs. *arXiv preprint arXiv:2305.06590*.

Lehmann, J.; Isele, R.; Jakob, M.; et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2): 167–195.

Li, G.; Wang, P.; and Ke, W. 2023. Revisiting Large Language Models as Zero-shot Relation Extractors. In *EMNLP*.

Li, J.; Cheng, X.; Zhao, X.; Nie, J.-Y.; and Wen, J.-R. 2023. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In *EMNLP*.

Liévin, V.; Hother, C. E.; and Winther, O. 2022. Can large language models reason about medical questions? *arXiv preprint arXiv:2207.08143*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; et al. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, J.; Wang, P.; Shang, Z.; and Wu, C. 2023. IterDE: an iterative knowledge distillation framework for knowledge graph embeddings. In *AAAI*.

Maynez, J.; Narayan, S.; Bohnet, B.; and McDonald, R. 2020. On faithfulness and factuality in abstractive summarization. In *ACL*.

McKenna, N.; Li, T.; Cheng, L.; Hosseini, M. J.; Johnson, M.; and Steedman, M. 2023. Sources of Hallucination by Large Language Models on Inference Tasks. *arXiv preprint arXiv:2305.14552*.

Min, S.; Krishna, K.; Lyu, X.; et al. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. *arXiv preprint arXiv:2305.14251*.

Mountantonakis, M.; and Tzitzikas, Y. 2023. Using Multiple RDF Knowledge Graphs for Enriching ChatGPT Responses. *arXiv preprint arXiv:2304.05774*.

Ouyang, L.; Wu, J.; Jiang, X.; et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Pan, J. Z.; Pavlova, S.; Li, C.; et al. 2018. Content based fake news detection using knowledge graphs. In *ISWC*.

Pellissier Tanon, T.; Weikum, G.; and Suchanek, F. 2020. Yago 4: A reason-able knowledge base. In *ESWC*.

Pezeshkpour, P. 2023. Measuring and Modifying Factual Knowledge in Large Language Models. *arXiv preprint arXiv:2306.06264*.

Qin, C.; Zhang, A.; Zhang, Z.; et al. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Radford, A.; Wu, J.; Child, R.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Sanh, V.; Webson, A.; Raffel, C.; et al. 2022. Multitask prompted training enables zero-shot task generalization. In *ICLR*.

Scao, T. L.; Fan, A.; Akiki, C.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Shang, Z.; Wang, P.; Liu, Y.; Liu, J.; and Ke, W. 2023. ASKRL: An Aligned-Spatial Knowledge Representation Learning Framework for Open-World Knowledge Graph. In *ISWC*.

Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S. S.; et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972): 172–180.

Studer, R.; Benjamins, V. R.; and Fensel, D. 1998. Knowledge engineering: Principles and methods. *Data & knowledge engineering*, 25(1-2): 161–197.

Touvron, H.; Lavril, T.; Izacard, G.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, S.; Wang, L.; and Mao, W. 2020. A KG-based Enhancement Framework for Fact Checking Using Category Information. In *ISI*.

Xie, Y.; Sun, F.; Deng, Y.; et al. 2021. Factual consistency evaluation for text summarization via counterfactual estimation. In *ACL*.

Xiong, H.; Wang, S.; Zhu, Y.; et al. 2023. Doctorglm: Finetuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*.

Xu, B.; Xu, Y.; Liang, J.; et al. 2017. CN-DBpedia: A never-ending Chinese knowledge extraction system. In *IEA/AIE*.

Yu, S.; Yuan, Z.; Xia, J.; et al. 2022. Bios: An algorithmically generated biomedical knowledge graph. *arXiv preprint arXiv:2203.09975*.

Zeng, A.; Liu, X.; Du, Z.; et al. 2023. Glm-130b: An open bilingual pre-trained model. In *ICLR*.

Zhang, M.; Press, O.; Merrill, W.; Liu, A.; and Smith, N. A. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Zhang, S.; Roller, S.; Goyal, N.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zheng, S.; Huang, J.; and Chang, K. C.-C. 2023. Why does chatgpt fall short in providing truthful answers. *ArXiv preprint, abs/2304.10513*.