SCIENCE CHINA
Information Sciences

• LETTER •

October 2024, Vol. 67, Iss. 10, 209101:1–209101:2
https://doi.org/10.1007/s11432-024-4158-4

Unveiling factuality and injecting knowledge for LLMs via reinforcement learning and data proportion

Wenjun KE[*†1,2], Ziyu SHANG[†1], Zhizhao LUO[3], Peng WANG[*1,2], Yikai GUO[4], Qi LIU[4] & Yuxuan CHEN[1]

[1]*School of Computer Science and Engineering, Southeast University, Nanjing* 210096*, China;*
[2]*Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Nanjing* 210096*, China;*
[3]*Beijing Institute of Technology Zhuhai, Zhuhai* 519088*, China;*
[4]*Beijing Institute of Computer Technology and Application, Beijing* 100048*, China*

Received 28 April 2024/Revised 18 July 2024/Accepted 3 September 2024/Published online 20 September 2024

Large language models (LLMs) have demonstrated remarkable effectiveness across various natural language processing (NLP) tasks, as evidenced by recent studies [1, 2]. However, these models often produce responses that conflict with reality due to the unreliable distribution of facts within their training data, which is particularly critical for applications requiring high credibility and accuracy [3]. It is necessary to enhance the factual accuracy of LLMs, which is paramount for their effective deployment in knowledge-intensive tasks [4]. Improving the reliability of these models could significantly advance their applicability and trustworthiness in real-world scenarios.

As shown in Figure 1, we propose a comprehensive approach with two key components: OntoFact and OntoInjection. OntoFact is an adaptive framework that uses an ontology-driven reinforcement learning (ORL) mechanism to detect unknown facts and extract ontology-level knowledge. It generates error-prone test cases and identifies unfactual triples within knowledge graphs (KGs), allowing precise detection of factual inaccuracies in LLMs. The ORL mechanism enables dynamic interaction between ontologies and instances, effectively predicting errors and compiling misleading ontologies into a knowledge skeleton.

Building upon the detection capability of OntoFact, OntoInjection aims to enhance the factual accuracy of LLMs by injecting missing knowledge. This method involves strategically adjusting data proportions to ensure high coverage of valuable data across various domains. OntoInjection initiates the process by web-crawling natural language sentences corresponding to the missing KG triplets. It then devises a novel homogeneity score metric to assess the alignment between LLMs and the retrieved sentences, selecting the most relevant and accurate samples for knowledge injection. The incremental fine-tuning mechanism, LoRA [5], is used to integrate this refined knowledge into LLMs.

OntoFact. It is an adaptive framework designed to detect factual inaccuracies in LLMs by leveraging an ORL mechanism. The process involves three primary stages.

(1) Test case initialization. In this stage, ontology-level triples (Ch, r, Ct) map to instance-level (h, r, t) for question templates. For example, an ontology-level triple $(Person, birthPlace, City)$ generates the question template: *"Was [Person] born in [City]?"*.

(2) Test case updation. The ORL mechanism guides the dynamic generation of test cases by exploring both ontology and instance views of KGs. The instance-view agent decides whether an instance-level triple (h, r, t) should be included as a test case using a policy function:

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

where $a_t^I \in \{0, 1\}$ is the action chosen by the agent at time step t, indicating whether the instance-level triple (h, r, t) is selected ($a_t^I = 1$) or not ($a_t^I = 0$). The state s_t^I represents the concatenation of embeddings corresponding to the current instance triple (h, r, t). The function $f_{\theta_I}(\cdot)$, implemented by a multi-layer perceptron, denotes the probability that the triple (h, r, t) is selected as a test case.

The instance-view agent receives rewards based on the consistency of the LLMs' answers with the actions, aiming to maximize the total reward $R(\tau)$:

$$R(\tau) = \sum_{t=0}^{|\tau|-1} \gamma^t r[t] + \alpha r', \quad (2)$$

where the term $\sum_{t=0}^{|\tau|-1} \gamma^t r[t]$ represents the discounted rewards at each time step t, with γ as the discount factor. The additional term $\alpha r'$ adjusts the return based on a global reward r', weighted by the hyperparameter α.

The ontology-view agent operates in a continuous space and updates actions using actor-critic networks to enhance the exploration of error-prone ontologies. The optimization of the critic network $Q_w(\cdot)$ is performed by minimizing the mean squared error (MSE) loss:

$$L(Q_w) = \frac{1}{|M|} \sum_i (y_i - Q_w(s_i^o, a_i^o))^2, \quad (3)$$

* Corresponding author (email: kewenjun@seu.edu.cn, pwang@seu.edu.cn)
† Wenjun KE and Ziyu SHANG have the same contribution to this work.

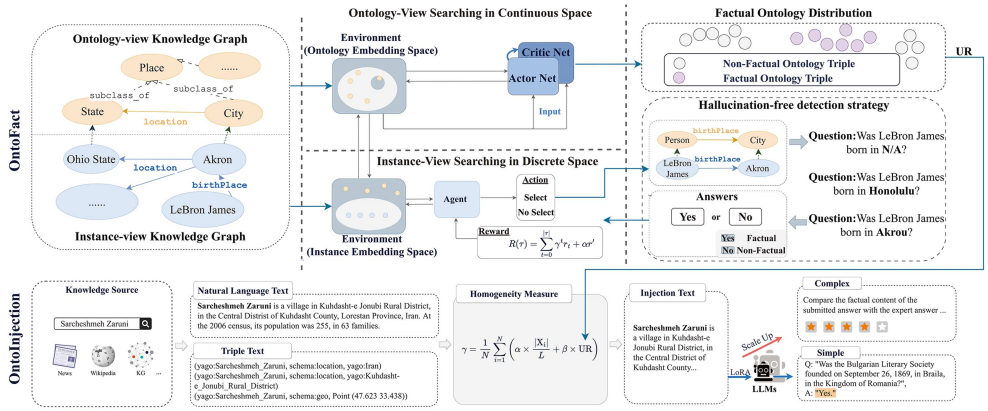

Figure 1 (Color online) Overview of OntoFact and OntoInjection, where arrows indicate the data flow from one stage to the next.

where $y_i = r'_i + \gamma Q_{\bar{w}}(s^o_{i+1}, \mu_{\bar{\theta}}(s^o_{i+1}))$. $L(Q_w)$ represents the loss function for the critic network Q_w, M is the number of samples used in the optimization process, y_i is the target value for the i-th sample, and $Q_{\bar{w}}(s^o_{i+1}, \mu_{\bar{\theta}}(s^o_{i+1}))$ represents the estimated value of the next state-action pair using the target critic and actor networks.

(3) Test case execution. A hallucination-free detection (HFD) strategy is employed to counteract LLMs' overconfidence. The strategy involves creating negative samples to test the LLMs' factuality.

$$\text{LLMs}(Q_E) = \chi[s_1 = \text{No}] \wedge \chi[s_2 = \text{No}] \wedge \chi[s_3 = \text{Yes}], \quad (4)$$

where $\chi[\cdot]$ denotes the indicator operator. Q_E represents unfactual test cases, while s_1, s_2, and s_3 denote LLMs' answer *w.r.t.* simple negative, hard negative, and positive samples.

Finally, we calculate the unfact rate (UR) for each ontology-level triple and gather those with UR greater than 50% into the unfactual ontology skeleton set.

OntoInjection. It builds on the results of OntoFact to enhance LLMs' factual accuracy by injecting missing knowledge. The process is divided into three key stages.

(1) Preliminary dataset construction. It involves creating datasets that support continual pre-training of LLMs. The top n ontology-level triples (Ch, r, Ct) with the highest UR are selected, and corresponding instance-level triples are used to generate triple texts (TTs) and natural language texts (NLTs).

(2) Knowledge filtering. It aims to filter high-quality NLTs, and the ontology density ρ and the degree of missing knowledge τ are calculated:

$$\rho = \frac{1}{N}\sum_{i=1}^{N}\frac{|X_i|}{L}, \tau = \frac{1}{N}\sum_{i=1}^{N}\text{UR}((Ch_i, r_i, Ct_i)), \quad (5)$$

where L is NLT length, N ontology-level triple count, and $|\text{X}_\text{i}|$ instance-level triple length for the i-th ontology-level triple. The homogeneity metric for ranking NLTs is:

$$\text{score} = \alpha \times \rho + \beta \times \tau, \quad (6)$$

where α and β are hyper-parameters.

(3) Data proportion for knowledge injection. It adjusts the composition of training dataset and uses LoRA to optimize knowledge injection objective function:

$$\begin{aligned} L(\theta_{\text{LLMs}}) &= \sum_{x \in D} -\log p_{\theta_{\text{LLMs}}}(x) \\ &= \sum_{x \in D}\sum_{i=1}^{L} -\log p_{\theta_{\text{LLMs}}}(x_i | x_{1:i-1}). \end{aligned} \quad (7)$$

In dataset D, $\{x_1, x_2, \cdots, x_L\}$ represents a text sequence, with L as its maximum length. The function $p_{\theta_{\text{LLMs}}}(\cdot,\cdot)$ calculates the probability of predicting the next word in the sequence. Extensive continual pre-training yields refined LLMs (θ'_{LLMs}) with improved factual accuracy. This knowledge injection is assessed through multiple-choice and open-ended question-answering paradigms.

Result. OntoFact evaluation across 32 LLMs reveals factual error rates of 50.1% (general domain) and 22.7% (biomedical), reducing error prediction time by 35.29%–63.12%. OntoInjection assessment on 5 LLMs shows fact rate improvements of 3.1%–56.6% using NLTs against TTs. NLTs filtered by homogeneity scores increase knowledge injection by 21.9%–57.4% compared to random text selection.

Conclusion. We propose a comprehensive framework for enhancing the factuality of LLMs through OntoFact and OntoInjection. OntoFact detects and highlights factual inaccuracies, while OntoInjection strategically injects the necessary knowledge. Our approach demonstrates significant improvements in LLM performance, paving the way for more reliable and accurate language models.

Acknowledgements This work was supported by the National Science Foundation of China (Grant No. 62376057).

References

1 Pan S, Luo L, Wang Y, et al. Unifying large language models and knowledge graphs: a roadmap. IEEE Trans Knowl Data Eng, 2024, 36: 3580–3599
2 Zeng A, Liu X, Du Z, et al. GLM-130B: an open bilingual pre-trained model. In: Proceedings of the International Conference on Learning Representations, 2023
3 Yao Y, Wang P, Tian B, et al. Editing large language models: problems, methods, and opportunities. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023. 10222–10240
4 Yang L, Chen H, Li Z, et al. Give us the facts: enhancing large language models with knowledge graphs for fact-aware language modeling. IEEE Trans Knowl Data Eng, 2024, 36: 3091–3110
5 Hu E J, Shen Y L, Wallis P, et al. LoRA: low-rank adaptation of large language models. In: Proceedings of the International Conference on Learning Representations, 2022