Knowledge Injection for Large Language Models

Anonymous ACL submission

Abstract

Generative Large Language Models (LLMs), such as ChatGPT and GPT-4, offer interactive APIs that can answer common questions at the human-expert level. However, these models often give inaccurate responses when faced with questions requiring domain-specific or professional-specific knowledge not covered in their training corpus. To alleviate this issue, Knowledge Graphs (KGs) have been integrated into LLMs as an additional source of knowledge. However, many state-of-the-art LLMs are not open-source, making it challenging to inject knowledge with model APIs only. In this paper, we propose a novel framework `KnowGPT`, which necessitates the *knowledge injection* for both knowledge retrieval and translation for LLMs. `KnowGPT` leverages (i) deep reinforcement learning to carefully extract context-aware knowledge from KGs, and (ii) a multi-armed bandit to construct an appropriate prompt format for each question. It significantly outperforms the existing methods on three benchmark datasets. Notably, `KnowGPT` attains a 91.6% accuracy on OpenbookQA official leaderboard, which is comparable to human performance. The code will be open-sourced.

1 Introduction

Large language models (LLMs) have surprised the world with their superior performance (Kung et al., 2023; Zha et al., 2023), especially with the emergence of ChatGPT and GPT-4 (OpenAI, 2023). Nonetheless, LLMs are often criticized for their limited factual knowledge and propensity to produce hallucinations, wherein the model fabricates incorrect statements on tasks beyond their knowledge and perception (Amaro et al., 2023; Shen et al., 2023; Gravel et al., 2023). Consider an ecological domain-specific question from OpenbookQA (Mihaylov et al., 2018) in Figure 1. ChatGPT erroneously responds "energy" when asked about the portion of nutrients. This inaccuracy

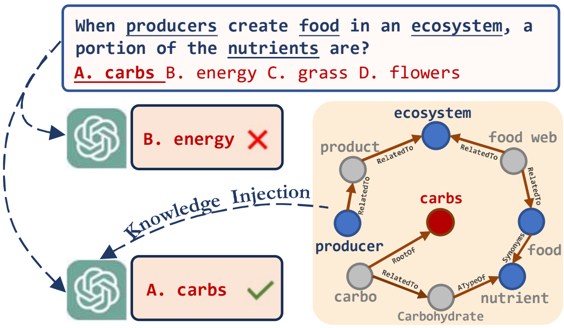

Figure 1: A real-world question in OpenbookQA and a subgraph of ConceptNet. ChatGPT could effectively correct the answer given the scientific reasoning background in KG (blue: question concepts, red: candidate answers).

could stem from its potential lack of knowledge of carbs and their relationship to nutrients.

A promising avenue for addressing the above issue entails the integration of Knowledge Graphs (KGs) into LLMs. KGs, such as Yago (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), and ConceptNet (Speer et al., 2017) represent relationships among real-world entities in a structured form as triples *(head, relation, tail)*. The enormous factual knowledge stored in KGs holds the potential to significantly enhance the accuracy of LLMs' responses. For instance, in Figure 1, ChatGPT could correct itself by leveraging the related background knowledge in ConceptNet (Speer et al., 2017).

Many algorithms have been proposed to integrate KGs into LLMs (Pan et al., 2023). Early studies directly concatenate the entities from KGs with textual sentences as the input to train LLMs based on cross-modal representation alignment (Sun et al., 2020b; Liu et al., 2020). Later research incorporates KGs at an implicit level by either combining text representation and KG embedding with the attention mechanism (Feng et al., 2020; Yasunaga et al., 2021; Lin et al., 2019; Dong et al., 2023a) or designing various fusion methods to integrate KGs and texts through a tailored encoder (Zhang et al., 2019; Sun et al., 2020a).

Unfortunately, many state-of-the-art LLMs are confined to be close-source in practical applications. For instance, ChatGPT and GPT-4 exclusively grant access through their APIs, which means we can only retrieve model responses by submitting textual inputs, with model specifics inaccessible. This lack of access prevents us from employing the aforementioned *white-box* knowledge injection techniques. Even though white-box approaches could be applied to open-source LLMs, such as BLOOM (Scao et al., 2022) and LLaMA (Touvron et al., 2023), they will often incur significant computation costs due to updating model weights (Liu et al., 2022). Thus, we ask: ***Can we develop a knowledge injection framework that can efficiently and effectively integrate KGs into LLMs with APIs only?***

Achieving this goal is nontrivial because of two challenges in constructing model inputs, or prompts. ❶ Identifying the most relevant knowledge is difficult. Real-world KGs often consist of millions of triples, whereas LLMs are typically restricted by limited input lengths (e.g., 2048 tokens for ChatGPT and 4096 tokens for GPT-4). Hence, careful selection of the most informative knowledge from KGs becomes essential. ❷ Effectively encoding KG knowledge is hard. It is observed that even minor variations in prompts conveying the same semantic meaning can yield drastically different responses from LLMs (OpenAI, 2023). As a result, a customized approach to encoding factual knowledge from extracted KGs for each question is often required to achieve the best performance.

In this work, we propose KnowGPT, a knowledge injection framework for LLMs in question answering. To address challenge ❶, we leverage deep reinforcement learning (RL) to extract paths from source entities mentioned in the question to the target entities within the potential answers. To encourage the agent to discover more informative paths, we devise a tailored reward scheme that promotes the reachability, context-relatedness, and conciseness of the extracted paths. Then, a policy network is trained to maximize the reward using training questions and applied to unseen questions. To tackle challenge ❷, we introduce a prompt construction strategy based on Multi-Armed Bandit (MAB). Given several path extraction strategies and prompt templates, a MAB is learned to select the most effective combination for each question by balancing exploration and exploitation. The learned MAB is then applied to new questions to select path extraction strategies and prompt templates automatically. Our main contributions are:

- Formally define the problem of *balck-box knowledge injection for LLMs*, which integrates KGs into LLMs with model APIs only.
- Propose KnowGPT, a general knowledge injection model to capture and translate knowledge from Knowledge Graphs (KGs) into prompts.
- Instantiate KnowGPT upon two real-world KGs, ConceptNet (Speer et al., 2017) and USMLE (Yasunaga et al., 2021), with ChatGPT APIs. KnowGPT outperforms the state-of-the-art baselines on three QA benchmarks by a large margin. Notably, KnowGPT attains a 91.6% accuracy on the OpenbookQA leaderboard, which is comparable to human performance.

2 Related Work

2.1 Integration of KGs and LLMs

Leveraging LLMs to assist KG-based tasks has been intensively studied. Colake (Sun et al., 2020b) presents a unified graph that combines a word graph with the given context and KG. QA-GNN (Yasunaga et al., 2021) takes this further and embeds the question context as an entity in the joint graph. Recent studies utilize the attention mechanism to incorporate KGs into LMs, thereby enhancing comprehension and reasoning processes. HamQA (Dong et al., 2023a) proposes a hyperbolic-based graph attention network to learn from the ubiquitous hyponymy in real-world questions. However, they are limited to adapt to LLMs since most existing LLMs are either closed-sourced or over-expensive to be effectively tuned.

To alleviate this issue, recent research has shown various ways to prompt the KG to LLMs. One heuristic way to incorporate KGs and LLMs is to inject triples as the input (Pan et al., 2023). ERNIE3.0 (Sun et al., 2021) takes triples as token sequences and straightforwardly appends them to the given sentences. Another group of work dedicate to designing various ways to retrieve the knowledge. Mindmap (Pan et al., 2024) transforms the relevant knowledge into mindmaps and prompt the LLMs. ChatRule (Luo et al., 2023) extracts multiple relational paths from KGs and prompts the LLMs with natural language sentences. CoK (Wang et al., 2023) activates the LLMs' comprehension through chain-of-knowledge which prompts triples step-by-step. However, they lack

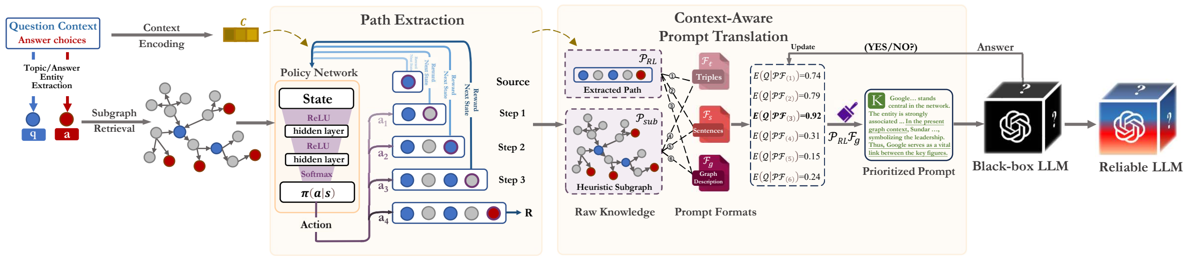

Figure 2: The overall architecture of KnowGPT. Given the question context and answer choices, we retrieve a question-specific subgraph from the real-world KG. *Path Extraction* is first dedicated to searching for the most informative and concise reasoning background subject to the context. Then the *prompt translation* module is optimized to prioritize the combination of knowledge and formats subject to the given question context.

the prompt format design which is laborious to be expanded to more LLMs and task scenarios. In this work, we develop a *knowledge injection* framework that values both knowledge retrieval and translation. It automatically prioritize relevant knowledge and prompts the LLMs with the the most suitable prompt formats.

3 Problem Statement

We formally define the problem of *black-box knowledge injection for LLMs* in complex question answering. We represent each question as a question context $\mathcal{Q} = \{\mathcal{Q}_s, \mathcal{Q}_t\}$, where $\mathcal{Q}_s = \{e_1, e_2, ..., e_m\}$ is a set of m *source entities*, and $\mathcal{Q}_t = \{e_1, e_2, ..., e_n\}$ is a set of n *target entities*. Following prior work (Feng et al., 2020; Yasunaga et al., 2022), $\mathcal{Q}_s$ is extracted by concept recognition, and we assume it is given in our problem. Similarly, each target entity in $\mathcal{Q}_t$ is extracted from a corresponding candidate answer. We denote an LLM as f, a real-world KG as $\mathcal{G}$, which consists of triples (head entity, relation, tail entity), denoted as (h, r, t). In our setting, we only have access to the APIs of f. However, we can employ open-source lightweight language models (not f), like Bert-Base (Kenton and Toutanova, 2019), to obtain text embeddings. Using the above notations, we describe our problem below.

> Given a question context $\mathcal{Q}$, an LLM f, and a KG $\mathcal{G}$, we aim to learn a prompting function $f_{\text{prompt}}(\mathcal{Q}, \mathcal{G})$, which generates a prompt $\mathbf{x}$ that incorporates the context of $\mathcal{Q}$ and the factual knowledge in $\mathcal{G}$, such that the prediction of the LLM $f(\mathbf{x})$ can output the correct answers for $\mathcal{Q}$.

4 KnowGPT Framework

Learning the prompting function $f_{\text{prompt}}(\mathcal{Q}, \mathcal{G})$ involves two challenges, i.e., what knowledge should be used in $\mathcal{G}$, and how to translate the structured knowledge into prompts. To address these challenges, we present KnowGPT, which extracts subgraphs (paths) with deep RL and then constructs the prompt with MAB. An overview of our framework is shown in Figure 2.

4.1 Knowledge Extraction with Deep Reinforcement Learning

Intuitively, the relevant reasoning background lies in a question-specific subgraph $\mathcal{G}_{\text{sub}}$ that contains all the *source* entities $\mathcal{Q}_s$, *target* entities $\mathcal{Q}_t$, and their neighbors. An ideal subgraph $\mathcal{G}_{\text{sub}}$ is expected to have the following properties: (i) $\mathcal{G}_{\text{sub}}$ encompasses as many source and target entities as possible, (ii) the entities and relations within $\mathcal{G}_{\text{sub}}$ exhibit a strong relevance to the question context, and (iii) $\mathcal{G}_{\text{sub}}$ is concise with little redundant information such that it can be fed into LLMs with limited input lengths.

However, it is challenging to find such a $\mathcal{G}_{\text{sub}}$ since extracting a subgraph is NP-hard. To effectively and efficiently find a satisfactory $\mathcal{G}_{\text{sub}}$, we develop a tailored path extraction method, named $\mathcal{P}_{\text{RL}}$, that employs deep RL to sample reasoning paths in a trial-and-error fashion. Specifically, we assume $\mathcal{G}_{\text{sub}}$ is constructed based on a set of reasoning paths $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_m\}$, where each path $\mathcal{P}_i = \{(e_i, r_1, t_1), (t_1, r_2, t_2), ..., (t_{|\mathcal{P}_i|-1}, r_{|\mathcal{P}_i|}, t_{|\mathcal{P}_i|})\}$ is a path in $\mathcal{G}$ starting from the i-th source entity in $\mathcal{Q}_s$, and $|\mathcal{P}_i|$ is the path length. $\mathcal{G}_{\text{sub}}$ encompasses all the entities and relations appeared in $\mathcal{P}$. We model the sampling of each reasoning path as a Markov Decision Process (MDP) with state, action, transition, and reward, defined as follows.

- **State:** A state indicates the current location in KG, i.e., one of the entities in KG. Specifically, it represents the spatial change from entity h to t. Inspired by the prior study (Xiong et al., 2017), we define the state vector $\boldsymbol{s}$ as:

$$s_t = (e_t, e_{target} - e_t), \quad (1)$$

where e_t and e_{target} are the embedding vectors of the current entity and the target entity. To get the initial node embeddings for entities extracted from the background KG, we adopt the approach proposed by the previous study (Feng et al., 2020). Specifically, we transform knowledge triples from the KG into sentences and feed them into pre-trained LM to get node embeddings.

- **Action:** The action space encompasses all the neighboring entities of the current entity, enabling the agent to explore the KG flexibly. By taking an action, the agent will move from the current entity to the chosen neighboring entity.

- **Transition:** The transition model P measures the probability of moving to a new state (s') given existing state (s) and the undertaken action (a). In KGs, the transition model takes on the form $\mathrm{P}(s'|s,a) = 1$ if s is directed to s' through action a; Otherwise, $\mathrm{P}(s'|s,a) = 0$.

- **Reward:** To determine the quality of the formed path, we define the reward based on reachability:

$$r_{reach} = \begin{cases} +1, & \textit{if target;} \\ -1, & \textit{otherwise,} \end{cases} \quad (2)$$

which represents whether the path eventually reaches the target within limited steps. Specifically, the agent receives a reward of $+1$ if it can attain the target within K actions. Otherwise, it will receive -1 as the reward.

Reaching a target entity is not our sole focus. To avoid overlong and rigmarole paths, we also design two auxiliary rewards to promote context relatedness and path conciseness.

4.1.1 Context-relatedness Auxiliary Reward

The key motivation is to encourage paths closely related to the given question context. Specifically, we evaluate the semantic relevance of a path $\mathcal{P}_i$ to the context $\mathcal{Q}$. Inspired by the prevailing study (Yasunaga et al., 2021), a fixed but well-trained matrix $\boldsymbol{W}$ is applied to map the path embedding $\boldsymbol{\mathcal{P}}$ to the same semantic space with context embedding $\boldsymbol{c}$. To this end, this auxiliary reward is formulated as:

$$r_{\text{cr}} = \frac{1}{|i|} \sum_{source}^{i} cos(\boldsymbol{W} \times \boldsymbol{\mathcal{P}}_i, \boldsymbol{c}), \quad (3)$$

where $\boldsymbol{c}$ is the embedding of context $\mathcal{Q}$ we obtained from a pre-trained LM (Kenton and Toutanova, 2019) and the embedding of path $\mathcal{P}_i$ is the average of the embeddings of all the entities and relations we have walked through till i, i.e., $Avg(\boldsymbol{e}_{source} + \boldsymbol{r}\boldsymbol{e}_1 ... + \boldsymbol{e}_i)$, where $i \leq length(\mathcal{P}_{target})$. This step-by-step reward scheme provides rewards before the target is reached.

4.1.2 Conciseness Auxiliary Reward

There are two additional significant challenges for the candidate reasoning background. (i) The natural limitation of black-box LLMs for over-long context understanding gives constrained budgets for prompts, where the extracted path is expected to be concise enough to ensure the full understanding by black-box LLMs. (ii) The prohibitive cost of calling black-box LLMs' API guides the prompt to be more concise. By limiting the step size, we encourage the policy to find as much valuable information as possible within the shortest path length.

Considering the inevitable homogeneity in the large-scale real-world KG constructed from the online corpus, each step in the final path is ideally a necessity. Specifically, we evaluate the conciseness of a path to reduce twists and turns on redundant entities, e.g., synonyms. Thus, the reward for the conciseness of a path $\mathcal{P}_i$ is formulated as follows.

$$r_{\text{cs}} = \frac{1}{|\mathcal{P}_i|}. \quad (4)$$

Finally, we use the trade-off parameters to balance the significance of each reward: $r_{\text{total}} = \lambda_1 r_{reach} + \lambda_2 r_{cr} + \lambda_3 r_{cs}.$, where λ_1, λ_2, and λ_3 are hyperparameters.

4.1.3 Training Policy Network

To solve the MDP defined above, a tailored policy network $\pi_\theta(s,a) = p(a|s;\theta)$ is trained to extract a reasoning path in the KG. We optimize the network with policy gradient (Xiong et al., 2017). The optimal policy navigates the agent from the source entity to the target entity while maximizing the accumulated rewards.

4.2 Structured Knowledge Prompts

In this subsection, we design a tailored prompt construction strategy based on Multi-Armed Bandit (MAB). The key idea is to learn to select the best path extraction and prompt templates at a meta-level. We will begin by outlining the overall strategy, followed by detailing its instantiation with two path extraction methodologies and three templates.

Suppose we have several path extraction strategies $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_m\}$ and several candidate formats $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$. Each path extraction strategy $\mathcal{P}_i$ is a method for selecting a subgraph given a question context, such as the RL-based strategy discussed above. Every prompt template $\mathcal{F}_j$ represents a transformation of the triples within the subgraph into a natural language prompt.

The prompt construction problem is to identify the best combination of $\mathcal{P}$ and $\mathcal{F}$ for a given question. We define the overall process of selection as a reward maximization problem, $\max \sum r_{pf}$, where r_{pf} is obtained as:

$$\sigma(f(\mathcal{PF}_{(i)})) = \begin{cases} 1 & \textit{if accurate;} \\ 0 & \textit{otherwise.} \end{cases} \tag{5}$$

Specifically, $\mathcal{PF}_{(i)}$, $i \in \{0, 1, \cdots, m \times n\}$ is one of the combination, and $r_{pf} \in \{0, 1\}$ indicates the performance of the output of LLM in answering the current question.

To capture the context-aware correlation between questions and different combinations of knowledge and prompt formats, we formulate the selection mechanism of MAB with an expectation function $E(\cdot)$. It adaptively measures the potential expectation of a combination for different questions.

$$E(\mathcal{Q}|\mathcal{PF}_{(i)}) = \boldsymbol{c} \times \boldsymbol{\alpha}_{(i)} + \beta_{(i)}. \tag{6}$$

Here, $\boldsymbol{c}$ represents the embedding of $\mathcal{Q}$. The vector $\boldsymbol{\alpha}(i)$ corresponds to a set of non-negative parameters associated with $\mathcal{PF}(i)$, which have been learned during the previous k-1 iterations. Additionally, $\beta_{(i)}$ stands for a balancing factor introducing noise according to a Gaussian distribution.

Empirically maximizing $\boldsymbol{c} \times \boldsymbol{\alpha}_i$ could encourage exploitation (Chen et al., 2019; Dong et al., 2023b) for the best combination, we could effectively update $\boldsymbol{\alpha}_{(i)}$ via modeling the correlations between the context embedding of the anchor question $\mathbf{c}_i$ and all the previously selected contexts $\mathbf{C}_{(i)}$ for particular combination $\mathcal{PF}_{(i)}$ in former k steps, and the rewards $\mathbf{r}_{pf}^{(i)}$ obtained from the selection of the current combination. Concretely, the $\boldsymbol{\beta}^{(b)}$ is updated as:

$$J(\mathbf{C}_{(i)}^{(k)}, \mathbf{r}_{pf}^{(i)(k)}) = \sum_{k=1}^{K} (\mathbf{r}_{pf}^{(i)(k)} - \mathbf{C}_{(i)}^{(k)} \boldsymbol{\alpha}^{(i)})^2 + \lambda^i \parallel \boldsymbol{\alpha}^{(i)} \parallel_2^2 .$$
$$\rightarrow \boldsymbol{\alpha}^{(i)} = \left((\mathbf{C}_{(i)}^{(k)})^\top \mathbf{C}_{(i)}^{(k)} + \lambda^i \mathbf{I} \right)^{-1} (\mathbf{C}_{(i)}^{(k)})^\top \mathbf{r}_{pf}^{(i)(k)}. \tag{7}$$

Here, J denotes the OLS training loss. $\mathbf{I} \in \mathbb{R}^{d \times d}$ is an identity matrix and λ^i is a regularization factor that controls the complexity of the model.

Similarly, in order to encourage exploration within less frequently selected pairings, we employ an upper confidence bound approach to balance exploration and exploitation. This is achieved through the introduction of the parameter $\beta^{(i)}$. Inspired by prevailing studies (Walsh et al., 2009; Dong et al., 2023b), we can derive the following exploration term $\beta^{(i)}$:

$$\beta^{(i)} = \quad \gamma \times \sqrt{\mathbf{c}_i \left((\mathbf{C}_{(i)}^{(k)})^\top \mathbf{C}_{(i)}^{(k)} + \lambda^i \mathbf{I} \right)^{-1} (\mathbf{c}_{(i)})^\top}, \tag{8}$$

where γ is a fixed constant, i.e., $\gamma = 1 + \sqrt{ln(2/\delta)/2}$.

When the model picks a combination with a large $\boldsymbol{c} \times \boldsymbol{\alpha}_i$, it signifies an exploitation process. Likewise, when the model selects a combination with larger $\beta^{(i)}$, this variance indicates an exploration process due to the model making fewer selections of the current combination. Thus, jointly maximizing $\boldsymbol{c} \times \boldsymbol{\alpha}_i + \beta_{(i)}$ could help us get rid of the dilemma of exploration and exploitation.

Consequently, our MAB design can leverage the feedback from the LLM to optimize the selection policy. By maximizing the expectation function $E(\cdot)$, it learns to balance the exploitation and exploration to prioritize the most promising prompts for specific question contexts.

4.2.1 Implementation

We implement the above MAB strategies with two path extraction strategies and three templates. Note that our MAB design is general and can be implemented with more path extraction strategies and prompt templates for better performance. The path extraction strategies include:

- $\mathcal{P}_{\text{RL}}$: The RL-based path extraction strategy presented in the previous subsection.

- $\mathcal{P}_{\text{sub}}$: A heuristic sub-graph extraction strategy that extracts a 2-hop subgraph around both the source and target entities. Detailed implementation can be found in Appendix A.1. Since RL is notoriously unstable (Sutton and Barto, 2018), we introduce $\mathcal{P}_{\text{sub}}$ as an alternative candidate strategy for the MAB selection, ensuring a fallback option if the RL-based approach does not perform well.

The prompt templates include:

- **Triples**, denoted as $\mathcal{F}_t$, are indeed the originally extracted knowledge and empirically tested that could be understood by the black-box LLMs, e.g., (*Sergey_Brin*, *founder_of*, *Google*),(*Sundar_Pichai*, *ceo_of*, *Google*), (*Google*, *is_a*, *High-tech Company*).

- **Sentences** is a following solution to transform the knowledge into a colloquial $\mathcal{F}_s$, e.g., '*Sergey Brin, who is a founder of Google, a high-tech company, has now passed the reigns to Sundar Pichai, who is currently serving as the CEO of the company.*'

- **Graph Description**, $\mathcal{F}_g$ prompts the LLM by treating the knowledge as a structured graph. We preprocess the extracted knowledge with black-box LLM itself to generate the description by highlighting the center entity, e.g., '*Google, a high-tech company, stands central in the network. The entity is strongly associated with significant individuals in the tech industry. Sergey Brin, one of the founders, established Google, underscoring its historical beginnings. In the present graph context, Sundar Pichai is recognized as the CEO of Google, symbolizing the company's current leadership. Thus, Google serves as a vital link between these key figures.*'

Considering two path extraction methods: $\mathcal{P}_{\text{sub}}$ and $\mathcal{P}_{\text{RL}}$, as well as three prompt translation methods: $\mathcal{F}_t$, $\mathcal{F}_s$ and $\mathcal{F}_g$, the MAB is trained to learn from the feedback from LLMs to prioritize the most appropriate combination among two extraction methods and three predefined prompt formats for different real-world question contexts, i.e., $\mathcal{PF} = \{(\mathcal{P}_{sub}\mathcal{F}_t), (\mathcal{P}_{sub}\mathcal{F}_s), (\mathcal{P}_{sub}\mathcal{F}_g), (\mathcal{P}_{RL}\mathcal{F}_t), (\mathcal{P}_{RL}\mathcal{F}_s), (\mathcal{P}_{RL}\mathcal{F}_g)\}$.

5 Experiments

We conduct extensive experiments to evaluate KnowGPT on three benchmark question-answering datasets, covering both commonsense and domain-specific QA. Our experiments are designed to answer the following research questions:

- **RQ1**: How does KnowGPT perform when compared with the state-of-the-art LLMs and KG-enhanced QA baselines?

- **RQ2**: Does the proposed MAB-based prompt construction strategy contribute to the performance?

- **RQ3**: Can KnowGPT solve complex reasoning tasks, and is KG helpful in this reasoning process?

5.1 Experimental Setup

5.1.1 QA Datasets

We evaluate KnowGPT on three QA datasets spanning two fields: CommonsenseQA (Talmor et al., 2019) and OpenBookQA (Mihaylov et al., 2018) serve as benchmarks for commonsense reasoning, while MedQA-USMLE (Jin et al., 2021) acts as a domain-specific QA benchmark. The statistics of these three datasets can be found in Table 5 in the Appendix.

5.1.2 State-of-the-art Baselines

We carefully select baseline models from four categories for a comprehensive evaluation.

LM + Fine-tuning We compare our method with vanilla fine-tuned LMs. Specifically, we choose Bert-base, Bert-large (Kenton and Toutanova, 2019), and RoBerta-large (Liu et al., 2019) as representative fine-tune LM methods. To conduct commonsense and biomedical QA, we fine-tune these three LMs via an additional linear layer.

KG-enhanced LM We have also implemented several recently released models for integrating KGs into question answering, which encompass MHGRN (Feng et al., 2020), QA-GNN (Yasunaga et al., 2021), HamQA (Dong et al., 2023a), JointLK (Sun et al., 2022), GreaseLM (Zhang et al., 2022) and GrapeQA (Taunk et al., 2023). To ensure a fair comparison, we implement these baselines with advanced language models that are optimized for particular datasets. Specifically, RoBerta-large (Liu et al., 2019) is used for CommenseQA, while AristoRoBERTa (Clark et al., 2020) is designated for OpenBookQA. For MedQA, we opt for the top-tier biomedical language model, SapBERT (Liu et al., 2021). Note that due to the white-box nature of these methods and their high computation overheads, it is infeasible to apply them to state-of-the-art LLMs, like ChatGPT and GPT-4.

LLM We include several representative generative LLMs, including ChatGLM, ChatGLM2, Baichuan-7B, InternLM, GPT-3, ChatGPT and GPT-4 as knowledge-agnostic alternatives. Specifically, we used the model 'text-davinci-002' provided by OpenAI as the implementation of GPT-3, and 'gpt-3.5-turbo' and 'gpt-4' as the implementations of ChatGPT and GPT-4, respectively (we have provided more implementation details of all

Table 1: Performance comparison among state-of-the-art baselines and KnowGPT on three benchmark datasets.

Catagory	**Model**	**CommonsenseQA**		**OpenBookQA**		**MedQA**	
		IHdev-Acc.	IHtest-Acc.	Dev-Acc.	Test-Acc.	Dev-Acc.	Test-Acc.
LM + Fine-tuning	Bert-base	0.573	0.535	0.588	0.566	0.359	0.344
	Bert-large	0.611	0.554	0.626	0.602	0.373	0.367
	RoBerta-large	0.731	0.687	0.668	0.648	0.369	0.361
KG-enhanced LM	MHGRN	0.745	0.713	0.786	0.806	-	-
	QA-GNN	0.765	0.733	0.836	0.828	0.394	0.381
	HamQA	0.769	0.739	0.858	0.846	0.396	0.385
	JointLK	0.777	0.744	0.864	0.856	0.411	0.403
	GreaseLM	0.785	0.742	0.857	0.848	0.400	0.385
	GrapeQA	0.782	0.749	0.849	0.824	0.401	0.395
LLM + Zero-shot	ChatGLM	0.473	0.469	0.352	0.360	0.346	0.366
	ChatGLM2	0.440	0.425	0.392	0.386	0.432	0.422
	Baichuan-7B	0.491	0.476	0.411	0.395	0.334	0.319
	InternLM	0.477	0.454	0.376	0.406	0.325	0.348
	GPT-3	0.539	0.520	0.420	0.482	0.312	0.289
	ChatGPT	0.735	0.710	0.598	0.600	0.484	0.487
	GPT-4	0.776	0.786	0.878	0.910	0.739	0.763
LLM + KG Prompting	CoK	0.759	0.739	0.835	0.869	0.706	0.722
	ChatRule	0.743	0.731	0.820	0.863	0.710	0.724
	Mindmap	0.789	0.784	0.851	0.882	0.747	0.751
Ours	KnowGPT	**0.827**	**0.818**	**0.900**	**0.924**	**0.776**	**0.781**
KnowGPT vs. ChatGPT	+ 23.7% (Avg.)	+ 9.2%	+ 10.8%	+ 31.2%	+ 32.4%	+ 29.2%	+ 29.4%
KnowGPT vs. GPT-4	+2.9% (Avg.)	+ 5.1%	+ 3.3%	+ 2.2%	+ 1.4%	+ 3.7%	+ 1.8%

*We used 'text-davinci-002' provided by OpenAI as the implementation of GPT-3, and 'gpt-3.5-turbo' for ChatGPT.
*The results compared with fine-tuning LLMs on CommonsenseQA are placed in Table 7 of Appendix.

LLMs in Appendix A.4). The question-answering task is conducted under the zero-shot setting with the question query from the test set as input.

LLM + KG Prompting To verify the effectiveness of our proposed prompting strategy used KnowGPT, we also add the state-of-the-art KG prompting methods, i.e., CoK (Wang et al., 2023), ChatRule (Luo et al., 2023), and Mindmap (Pan et al., 2024) as baselines.

Table 2: OpenBookQA Official Leaderboard records of three groups of related models (sorted by rankings).

OpenBookQA Leaderboard	
Human Performance	**0.917**
w/o KG	0.778
MHGRN (Feng et al., 2020)	0.806
QA-GNN (Yasunaga et al., 2021)	0.828
GreaseLM (Zhang et al., 2022)	0.848
HamQA (Dong et al., 2023a)	0.850
JointLK (Sun et al., 2022)	0.856
GSC (Wang et al., 2021)	0.874
UnifiedQA (Khashabi et al., 2020)	0.872
DRAGON (Yasunaga et al., 2022)	0.878
GenMC (Huang et al., 2022)	0.898
GenMC Ensemble (Huang et al., 2022)	0.920
MVP-Tuning Ensemble (Huang et al., 2023)	0.952
KnowGPT	**0.916**

5.2 Main Results (RQ1)

To address **RQ1**, we evaluate KnowGPT by comparing it to state-of-the-art baselines on the three benchmark datasets. KnowGPT is based on the original ChatGPT. We measure the performance using accuracy, which calculates the percentage of questions correctly predicted by the model out of the total questions in the test set. We have the following observations:

- KnowGPT outperforms all categories of methods, including sixteen different baselines, across all datasets and model architectures. This suggests that KnowGPT can effectively inject the knowledge from KGs to LLMs.

- KnowGPT surpasses the performance of ChatGPT and even GPT-4. On average, KnowGPT achieves a 23.7% higher testing accuracy than ChatGPT. Specifically, KnowGPT outperforms ChatGPT by 10.8%, 32.4%, and 29.4% on the CommonsenseQA, OpenBookQA, and MedQA datasets, respectively. More importantly, despite being based on ChatGPT, KnowGPT outperforms the state-of-the-art LLM GPT-4 by 3.3%, 1.4%, and 1.8% on the CommonsenseQA, OpenBookQA, and MedQA datasets, respectively. These results confirm that black-box knowledge injecting can effectively enhance the capabilities of LLMs.

- KnowGPT outperforms all KG-enhanced LMs significantly. This implies our black-box knowledge injection method proficiently encodes knowledge into LLMs. Furthermore, it showcases the superiority of our black-box approach, given its adaptable application to ChatGPT using only the model API, a feat not achievable by white-box methods.

5.2.1 Leaderboard Ranking

We submit our results onto the official leaderboard maintained by the authors of OpenbookQA. The full records on the leaderboard are shown on the website[1], while our result can be found from here[2].

We summarize the related submissions in Table 2, including three categories: traditional KG-enhanced LM, fine-tuning of LLMs, e.g., T5-11B used in UnifiedQA, and ensemble of multiple predictors. KnowGPT significantly outperforms traditional KG-enhanced LMs with 4.2% improvements when compared to the best baseline. Despite the 2.6% difference of the state-of-the-art fine-tuning method X-Reasoner, which is the currently best single model, they costly train the T5-11B LLM and also initialize the model on eight A100 GPUs. The third group of methods occupies the leaderboard by leveraging ensemble learning strategies. Nevertheless, KnowGPT can still obtain competitive performance without ensembling with merely 0.4% below GenMC Ensemble (Huang et al., 2022). Notably, our KnowGPT is remarkably comparable to the human performance.

Table 3: Ablation study on the effectiveness of two path extraction methods.

Path Extraction	Model	CSQA IHdev	CSQA IHtest	OBQA Test	MedQA Test
w/o KG	GPT-3	0.539	0.520	0.482	0.289
	ChatGPT	0.735	0.710	0.598	0.487
	GPT-4	0.776	0.786	0.910	0.763
$\mathcal{P}_{\text{sub}}$	ChatGPT	0.750	0.739	0.865	0.695
$\mathcal{P}_{\text{RL}}$	ChatGPT	0.815	0.800	0.889	0.755
Ours	KnowGPT	**0.827**	**0.818**	**0.924**	**0.781**

5.3 Ablation Studies (RQ2)

To answer **RQ2**, we conduct two ablation studies. **First**, in Table 3, we measure the importance of the tailored reinforcement learning-based path extraction module, i.e., $\mathcal{P}_{\text{RL}}$. Specifically, we compare it with the heuristic sub-graph extraction strategy, i.e., $\mathcal{P}_{\text{sub}}$. The performance is evaluated by directly feeding the extracted knowledge with the prompt format of 'Sentence', i.e., $\mathcal{F}_s$, to ChatGPT. We also include 'w/o KG' as the baseline where ChatGPT is asked to independently answer the given question with no reasoning background provided. The results clearly indicate the vital role of our proposed path extraction strategies. **Second**, we compare each of the three prompt formats subject to the same extracted knowledge. The detailed results are shown in Table 4. Though different formats perform similarly within the difference of 2.2% - 3.3%, they are particularly suitable for different kinds of questions. We illustrate this observation in the following case study section. Both ablation studies support the indispensability of each module, armed with a tailored deep reinforcement learning-based path extraction and a context-aware prompt translation, our KnowGPT performs best on all three benchmark datasets.

Table 4: Ablation study on different prompt translation formats for the extracted knowledge.

Path Extraction	Prompts	CSQA IHdev	CSQA IHtest	OBQA Test	MedQA Test
$\mathcal{P}_{\text{sub}}$	$\mathcal{F}_t$	0.728	0.701	0.832	0.589
	$\mathcal{F}_s$	0.750	0.739	0.865	0.695
	$\mathcal{F}_g$	0.737	0.715	0.871	0.680
$\mathcal{P}_{\text{RL}}$	$\mathcal{F}_t$	0.782	0.769	0.853	0.739
	$\mathcal{F}_s$	0.815	0.800	0.889	0.755
	$\mathcal{F}_g$	0.806	0.793	0.906	0.762
Full KnowGPT		**0.827**	**0.818**	**0.924**	**0.781**

6 Conclusion and Future Work

In this work, we formally define the problem of *knowledge injection for LLMs* in complex question answering. A novel framework, namely KnowGPT, is presented to integrate KGs into LLMs effectively with model APIs only. We first train a deep RL policy to extract informative and concise reasoning paths from the KG. Then we learn an MAB to select the most suitable and effective path extraction method and prompt template subject to each question. Extensive experiments on both general and domain-specific QA datasets show the superior performance of KnowGPT and the effectiveness of each component. In the future, we will study more advanced path extraction strategies and prompt templates to improve the performance further.

[1] https://leaderboard.allenai.org/open_book_qa/submissions/public.

[2] https://leaderboard.allenai.org/open_book_qa/submission/cj9game4arcuacugbrj0.

Limitations

Through our exploration, we realize the natural limitations of KnowGPT brought by real-world KGs. Existing KGs are automatically constructed based on online corpora. This inevitably introduces a considerable number of noisy triples into KGs. Under such circumstances, the noisy knowledge may mislead the LLMs to wrong predictions despite the effectiveness of our injection methods. To apply KnowGPT to practical scenarios, we would leverage off-the-shelf KG refinement algorithms to improve the quality of KGs.

Ethics Statement

In this study, all the datasets are publicly available and have been extensively used in research related to neural language processing.

References

Ilaria Amaro, Attilio Della Greca, Rita Francese, Genoveffa Tortora, and Cesare Tucci. 2023. Ai unreliable answers: A case study on chatgpt. In *ICHCI*, pages 23–40. Springer.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877–1901.

Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. 2019. A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In *COLT*. PMLR.

Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, et al. 2020. From 'f'to 'a'on the ny regents science exams: An overview of the aristo project. *AI Magazine*, 41(4):39–53.

Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. 2023a. Hierarchy-aware multi-hop question answering over knowledge graphs. In *WWW*, pages 2519–2527.

Junnan Dong, Qinggang Zhang, Xiao Huang, Qiaoyu Tan, Daochen Zha, and Zhao Zihao. 2023b. Active ensemble learning for knowledge graph error detection. In *WSDM*, pages 877–885.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*, pages 1295–1309.

Jocelyn Gravel, Madeleine D'Amours-Gravel, and Esli Osmanlliu. 2023. Learning to fake it: limited responses and fabricated references provided by chatgpt for medical questions. *Mayo Clinic Proceedings: Digital Health*, 1(3):226–234.

Yongfeng Huang, Yanyang Li, Yichong Xu, Lin Zhang, Ruyi Gan, Jiaxing Zhang, and Liwei Wang. 2023. Mvp-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13417–13432.

Zixian Huang, Ao Wu, Jiaying Zhou, Yu Gu, Yue Zhao, and Gong Cheng. 2022. Clues before answers: Generation-enhanced multiple-choice qa. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3272–3287.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *EMNLP 2020*, pages 1896–1907.

Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP-IJCNLP*, pages 2829–2839.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *ACL*, pages 61–68.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, pages 2381–2391.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS*, 35:27730–27744.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. *arXiv preprint arXiv:2304.08979*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. 2020a. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020b. Colake: Contextualized language and knowledge embedding. *arXiv preprint arXiv:2010.00309*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5049–5060.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. 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.

Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. 2023. Grapeqa: Graph augmentation and pruning to enhance question-answering. In *Companion Proceedings of the ACM Web Conference 2023*, pages 1138–1144.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Thomas J Walsh, István Szita, Carlos Diuk, and Michael L Littman. 2009. Exploring compact reinforcement-learning representations with linear regression. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 591–598.

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. 2023. Boosting language models reasoning with chain-of-knowledge prompting. *arXiv preprint arXiv:2306.06427*.

Kuan Wang, Yuyu Zhang, Diyi Yang, Le Song, and Tao Qin. 2021. Gnn is a counter? revisiting gnn for question answering.

David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. 2017. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research*, 46:D1074–D1082.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *EMNLP*, pages 564–573.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *NAACL*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. *arXiv preprint arXiv:2201.08860*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.

A Case Studies (RQ3)

For **RQ3**, we provide insights into how KnowGPT facilitates the prompt translation with a real case from CommonsenseQA. We visualize both the extracted knowledge and the textual inputs to ChatGPT in Figure 3. In this example, given the same extracted knowledge, ChatGPT answers correctly based on the sentence format that we provide. In contrast, it fails to answer the question with triples and graph descriptions. They clearly indicate the superiority of KnowGPT in an automatic context-aware prompt translation. We make the following observations: (i) Triple format $\mathcal{F}_t$ is intuitively suitable for all the simple questions by directly indicating the one-hop knowledge. (ii) Graph description may inevitably introduce noise to ensure the completeness and contextual fluency of the directed graph. In this example, since 'vacation' appears in both question and answer choices, over-emphasizing and connecting the knowledge about 'vacation' with other concepts in the graph misleads the model to make a prediction with an oblique focus. (iii) Our KnowGPT has shown superior performance in automatically constructing suitable prompts for particular questions.

B Implementation Details

B.1 Entity Linking and Heuristic Path Extraction $\mathcal{P}_{sub}$

For each QA context, we adopt the methodology outlined in the prior research (Lin et al., 2019; Yasunaga et al., 2022) to extract the subgraph from the background knowledge graph (KG), denoted as $\mathcal{G}$. We commence by executing entity linking on $\mathcal{G}$, resulting in an initial collection of nodes, V_{topic}. Next, we incorporate bridge entities that appear within a 2-hop path between any two linked entities from V_{topic}, yielding the set $V_{retrieval}$. Subsequently, we refine this set by evaluating the relevance score for each node, adhering to the method proposed (Yasunaga et al., 2022). From this refined set, only the top 200 nodes, based on score, are retained, and the others are discarded. We then extract all edges connecting any pair of nodes in V_{sub}, creating the retrieved subgraph G_{sub}. Each node within G_{sub} is designated a type based on its association to either the topic entities Q or target entities A. Intuitively, the relevant reasoning background lies in a question-specific subgraph $\mathcal{G}_{sub}$ that contains all the *source* entities S, *target* entities A, and their k-hop neighbors. Therefore, the reasoning background could be provided as the $\mathcal{G}_{sub}$, we denote this direct path extraction method as $\mathcal{P}_{sub}$.

B.2 Graph Initialization

To calculate the initial node embeddings for entities extracted from the background KG, we adopt the approach proposed by the previous study (Feng et al., 2020). Specifically, we transform knowledge triples from the KG into sentences and feed them into pre-trained LMs to get node embeddings. Specifically, to ensure a fair comparison, we implement all the KG-enhanced baselines and our model with the same advanced language models that are optimized for particular datasets. Specifically, RoBert-large (Liu et al., 2019) is used for CommenseQA, while AristoRoBERTa (Clark et al., 2020) is designated for OpenBookQA. For MedQA, we opt for the top-tier biomedical language model, SapBERT (Liu et al., 2021), to enhance comprehension of the biomedical field.

B.3 Datasets

We evaluate KnowGPT on three QA datasets spanning two fields: CommonsenseQA (Talmor et al., 2019) and OpenBookQA (Mihaylov et al., 2018) serve as benchmarks for commonsense reasoning, while MedQA-USMLE (Jin et al., 2021) acts as a domain-specific QA benchmark. While the official test set serves primarily for leaderboard rankings, we initially assess model efficacy using the in-house (IH) data split introduced in (Lin et al., 2019). The official dataset is denoted as CSQA, while the IH split is represented by CSQA(IH)*. The statistics of these three datasets can be found in Table 5.

Table 5: The statistical information of three datasets.

Dataset	Question	Choices	Train	Dev	Test
CSQA	#12102	5	9741	1221	1140
CSQA(IH)	#12102	5	8500	1221	1241
OBQA	#5957	4	4957	500	500
MedQA	#12723	4	10178	1272	1273

CommonsenseQA is a multiple-choice question-answering dataset, each question accompanied by five potential answers. Answering its 12,102 questions necessitates a foundation in commonsense knowledge. While the official test set serves primarily for leaderboard rankings, we initially assess model efficacy using the in-house (IH) data split introduced in (Lin et al., 2019). The official dataset is denoted as CSQA, while the IH split is represented by CSQA(IH)*.

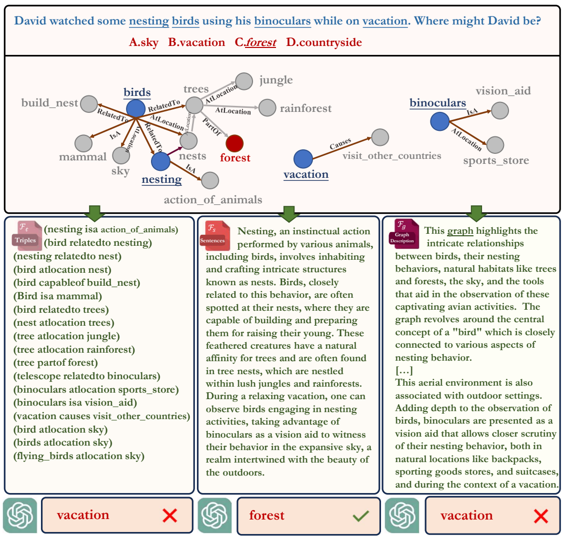

Figure 3: A case study on exploring the effectiveness of different prompt formats for particular questions. The extracted knowledge is shown in the middle of this figure in the form of a graph, where the nodes in blue are the key topic entities and the red is the target answer. The text boxes at the bottom are the final prompts generated based on three different formats.

OpenBookQA, commonly abbreviated as *OBQA*, comprises 5,957 multiple-choice questions, each offering four possible answers. To successfully answer these questions, one must have a comprehensive understanding of fundamental scientific facts and its applications.

MedQA-USMLE, abbreviated as *MedQA*, is a dataset consisting of 4-option multiple-choice questions that demand a grasp of biomedical and clinical understanding. These questions are sourced from preparatory tests for the United States Medical Licensing Examinations, and the dataset encompasses 12,723 questions. We adhere to the original data divisions as outlined in (Jin et al., 2021).

Background Knowledge To facilitate common sense reasoning, we employ **ConceptNet** (Speer et al., 2017), an extensive commonsense knowledge graph comprising more than 8 million interconnected entities through 34 concise relationships. For tasks specific to the medical domain, we leverage **USMLE** (Yasunaga et al., 2021) as our foundational knowledge source. USMLE is a biomedical knowledge graph that amalgamates the Disease Database segment of the Unified Medical Language System (UMLS) (Bodenreider, 2004) and DrugBank (Wishart et al., 2017). This repository encompasses 9,958 nodes and 44,561 edges.

B.4 Implementation of Baselines

To verify the effectiveness of our proposed KnowGPT, we carefully selected baseline models from three aspects to ensure a comprehensive evaluation, among which Bert-base, Bert-large (Kenton and Toutanova, 2019), and RoBert-large (Liu et al., 2019) are picked for being representative fine-tune LM methods; MHGRN (Feng et al., 2020), QA-GNN (Yasunaga et al., 2021), HamQA (Dong et al., 2023a), JointLK (Sun et al., 2022), GreaseLM (Zhang et al., 2022) and GrapeQA (Taunk et al., 2023) represent the state-of-art KG-enhanced LMs; ChatGLM (Du et al., 2022), ChatGLM2 (Zeng et al., 2022), Baichuan-7B, InternLM (Team, 2023), GPT-3 (Brown et al., 2020), ChatGPT (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023) are picked for being representative generative large language models. Notably, while some LLM baselines are actually open-source, we conduct the question-answering task under the zero-shot setting with the question query from the test set as input. All baseline methods used in this paper are based on their open-source implementations or officially-released APIs. We list the source links in Table 6. Notably, we used the model 'text-davinci-002' provided by OpenAI as the implementation of GPT-3, and 'gpt-3.5-turbo' and 'gpt-4' as the implementations of ChatGPT and GPT-4, respectively.

Table 6: Implementation codes for baselines.

Baseline	Code source
MHGRN	https://github.com/INK-USC/MHGRN.git
QA-GNN	https://github.com/michiyasunaga/qagnn.git
HamQA	https://github.com/DEEP-PolyU/HamQA_TheWebConf23.git
GreaseLM	https://github.com/snap-stanford/GreaseLM.git
JointLK	https://github.com/Yueqing-Sun/JointLK.git
ChatGLM	https://github.com/THUDM/ChatGLM-6B.git
ChatGLM2	https://github.com/THUDM/ChatGLM2-6B.git
Baichuan-7B	https://github.com/baichuan-inc/Baichuan-7B.git
InternLM	https://github.com/InternLM/InternLM.git
GPT3	https://platform.openai.com/docs/api-reference/models
ChatGPT	https://openai.com/chatgpt
GPT-4	https://openai.com/research/gpt-4
ChatRule	https://github.com/RManLuo/ChatRule

C Subplementary Results

C.1 Fine-tune LLMs

To further verify the effectiveness of the knowledge injection framework, we also add several open-source trainable LLMs, i.e., ChatGLM, ChatGLM2, LLaMA-B, Baichuan-7B, InternLM and Vicuna-7B, and fine-tune them for commonsense reasoning on the benchmark CommonseQA. As shown in the

Table 7, Our KnowGPT achieves comparable performance with no tuning on the LLM.

Table 7: Fine-tuned LLMs on CommonsenseQA.

LLM	Acc
ChatGLM	0.559
ChatGLM2	0.600
LLaMA-7B	0.650
Baichuan-7B	0.588
Alpaca-7B	0.687
Vicuna-7B	0.667
InternLM-7B	0.752
KnowGPT	**0.818**

C.2 The effect of prompt format on different types of questions.

In this part, we conduct comprehensive experiments to investigate the effect of prompt format on different types of questions. Table 8 presents the accuracy of different prompt formats on three types of questions. We observe that graph description-based prompt performs significantly better than any other prompt formats on complex questions. It is because graph description-based prompt could provide LLMs with more detailed and structured information by highlighting the local structure of the central entity.

Table 8: Accuracy of different prompt formats on specific types of questions on CommonsenseQA.

Prompt Format	Simple	Multi-hop	Graph reasoning
$\mathcal{F}_t$ (Triple)	**0.9412**	0.7447	0.4210
$\mathcal{F}_s$ (Sentence)	0.8823	**0.8298**	0.4739
$\mathcal{F}_g$ (Graph)	0.8529	0.7021	**0.7895**