Unifying Local and Global Knowledge: Empowering Large Language Models as Political Experts with Knowledge Graphs

Xinyi Mou
Fudan University
Shanghai, China
xymou20@fudan.edu.cn

Zejun Li
Fudan University
Shanghai, China
zejunli20@fudan.edu.cn

Hanjia Lyu
University of Rochester
Rochester, NY, USA
hlyu5@ur.rochester.edu

Jiebo Luo
University of Rochester
Rochester, NY, USA
jluo@cs.rochester.edu

Zhongyu Wei*
Fudan University
Shanghai, China
zywei@fudan.edu.cn

ABSTRACT

Large Language Models (LLMs) have revolutionized solutions for general natural language processing (NLP) tasks. However, deploying these models in specific domains still faces challenges like hallucination. While existing knowledge graph retrieval-based approaches offer partial solutions, they cannot be well adapted to the political domain. On one hand, existing generic knowledge graphs lack vital political context, hindering deductions for practical tasks. On the other hand, the nature of political questions often renders the direct facts elusive, necessitating deeper aggregation and comprehension of retrieved evidence. To address these challenges, we propose a **P**olitical **E**xperts through Knowledge **G**raph Integration (**PEG**) framework. PEG entails the creation and utilization of a multi-view political knowledge graph (MVPKG), which integrates U.S. legislative, election, and diplomatic data, as well as conceptual knowledge from Wikidata. With MVPKG as its foundation, PEG enhances existing methods through knowledge acquisition, aggregation, and injection. This process begins with refining evidence through semantic filtering, followed by its aggregation into global knowledge via implicit or explicit methods. The integrated knowledge is then utilized by LLMs through prompts. Experiments on three real-world datasets across diverse LLMs confirm PEG's superiority in tackling political modeling tasks.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; • **Human-centered computing** → *Collaborative and social computing*.

KEYWORDS

large language models, knowledge graph, political science

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW2024, May 13–17, 2024, Singapore

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3645616

ACM Reference Format:
Xinyi Mou, Zejun Li, Hanjia Lyu, Jiebo Luo, and Zhongyu Wei. 2024. Unifying Local and Global Knowledge: Empowering Large Language Models as Political Experts with Knowledge Graphs. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3589334.3645616

1 INTRODUCTION

Large Language Models (LLMs) have exhibited an impressive ability to tackle a wide range of tasks. As the scale of LLMs continues to expand, they possess the ability to answer questions based on their inherent knowledge, eliminating the need for additional fine-tuning [6, 30]. Nevertheless, when deployed in specific domains, these models still encounter certain challenges. Since intrinsic knowledge can be incomplete and outdated, LLMs may refuse to respond to a question or produce factually incorrect answers, leading to the well-known *hallucination* phenomenon [41]. This issue is particularly evident in the political domain, as shown in Figure 1(a), where LLMs struggle to perform tasks like political actor modeling and opinion mining without external knowledge.

To compensate for the knowledge gap of LLMs, a line of research proposes retrieval-based methods to augment LLMs via contextually relevant external knowledge. Early work [16, 19, 26, 42] utilizes documents as the sources of knowledge. Compared to documents, knowledge graphs (KGs) consisting of triples, *i.e.*, {(head entity, relation, tail entity)}, provide brief and explicit structural knowledge and explainable reasoning paths [52]. An additional advantage of KGs lies in their adaptability and expansibility, allowing for seamless modifications and additions. Considering this, some efforts [3, 43, 52] were made recently, to prompt LLMs to answer questions that can be resolved by referencing KGs. This is accomplished by providing LLMs with plain text, reformatted paths, or mindmaps that contain basic triples extracted from KGs.

Despite the remarkable achievements in general knowledge graph question answering (KGQA), existing approaches enhancing LLMs with KGs fall short in addressing challenges specific to the political domain. This limitation stems from several key factors: (1) **Knowledge-Task Mismatch.** Existing knowledge graphs, such as Wikidata [46] and Freebase [4], primarily contain generic information such as politicians' names and nationalities. However, it is inadequate for capturing the opinions and stances of politicians on specific policies - a critical aspect that attracts more attention in

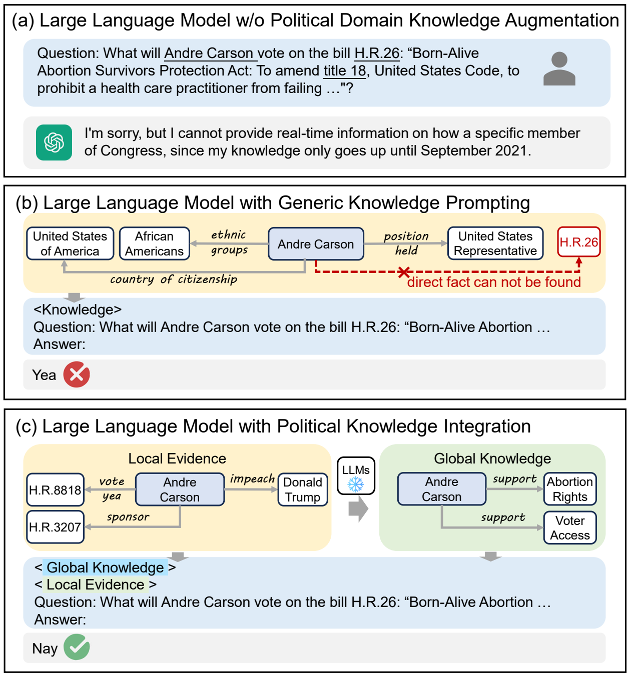

Figure 1: (a) An illustration of GPT-3.5 **[38] answering a roll call vote prediction question. It declined to answer due to outdated internal knowledge limitations. (b) Prompting LLMs with generic knowledge from Wikidata [46]. Existing solutions [3] cannot find the direct fact and generate the wrong answer. (c) Our PEG framework retrieves local evidence from our political domain KG, derives global knowledge, and unifies both knowledge, to generate the correct answer.**

the political domain. The mere inclusion of this basic knowledge into LLMs is insufficient for generating accurate responses due to the inherent mismatch between the available knowledge and the depth of inquiry within the political domain; (2) **Ineffective Direct Fact Retrieval.** Even if we enrich the current knowledge graphs with politicians' historical opinions about specific policies, existing methods for direct fact retrieval [2] may still encounter difficulties when applied to prediction tasks like vote prediction and event prediction, as the future fact can not be directly retrieved in KGs. As shown in Figure 1(b), existing KG-enhanced LLM frameworks like KAPING [3] are not able to produce accurate answers because they struggle to locate the relevant facts or construct the expected reasoning paths; (3) **Lack of Semantic Understanding.** The current approaches typically focus on presenting local evidence while neglecting the nuanced semantic relationships between pieces of evidence and their derived high-level contextual clues, resulting in an incomplete comprehension of the acquired knowledge.

In light of these challenges, we propose to enhance Large Language Models as **P**olitical **E**xperts through Knowledge **G**raph Integration (**PEG**). This framework leverages political domain knowledge to incorporate background information and augment LLMs in computational political tasks, comprising two key components:

First, to address the knowledge-task mismatch issue, we start with constructing a **multi-view political knowledge graph**, covering factual knowledge pertaining to U.S. politics, including legislation, election, and diplomatic events. This knowledge graph supplements generic conceptual knowledge by providing a tailored foundation for political expertise.

Second, based on this KG, we augment LLMs' inference by **knowledge acquisition, aggregation, and injection**. In particular, for each question, we extract relevant entities and explore their associated facts as candidate evidence. We then filter the evidence according to their semantic similarity to the question to reduce noise. This process intends to effectively retrieve relevant knowledge. After acquiring local evidence, we proceed to aggregate the local evidence into global knowledge, either implicitly through embedding techniques or explicitly making use of the strong summarization and reasoning capabilities of LLMs. Finally, we incorporate both the local evidence and global knowledge along with the question through pre-defined prompt templates, to guide LLMs in producing answers grounded in the provided knowledge with semantic understanding. As shown in Figure 1, **PEG** harnesses the cognitive capabilities and reasoning prowess of LLMs to consolidate localized evidence into a comprehensive body of contextual knowledge. This approach enables LLMs to deliver answers with greater depth, as exemplified in the case of attitudes on abortion issues. Here, LLMs can navigate a succinct reasoning path, *{H.R.26->against->abortion rights, Andre Carson->support->abortion rights}=>Andre Carson->against->H.R.26*, thereby eliminating the need for complex reasoning based solely on local evidence.

Our main contributions can be summarized as follows:

- To enhance large language models as political experts, we specifically construct a domain-specific political knowledge graph involving contemporary U.S. political facts of multiple perspectives, which consists of 116,176 entities, 602 relations, and 1,857,410 triples, publicly available at: https://github.com/xymou/PEG.
- We introduce a novel approach to mining high-level knowledge from localized facts, thus addressing situations where direct answers within knowledge graphs prove elusive. We provide both implicit and explicit implementations for different types of LLMs.
- We have conducted comprehensive experiments on diverse real-world datasets and across different LLMs. Our proposed methodology consistently demonstrates competitive performance in comparison to established baselines. Furthermore, a thorough analysis confirms the superior performance and interpretability of our approach.

2 MVPKG: A MULTI-VIEW POLITICAL KNOWLEDGE GRAPH

While existing generic knowledge graphs such as Wikidata KG [46], FreeBase [4] and YAGO [40] have proven valuable in a range of NLP tasks, their utility is largely confined to addressing basic demographic queries. These knowledge graphs, however, lack the capability to effectively support complex tasks related to political actor modeling and argumentative reasoning in politics-related tasks. To our knowledge, merely a single attempt [11] has been

Table 1: Examples of conceptual knowledge and factual knowledge in the MVPKG. Conceptual knowledge mainly presents the basic attributes of a person entity, while factual knowledge encompasses records of behaviors or events.

Conceptual Knowledge Examples	Factual Knowledge Examples
(Donald Trump, occupation, real estate entrepreneur)	(Andre Carson, sponsor bill, Patient Advocate Tracker Act…)
(Donald Trump, member of political party, Republican Party)	(Andre Carson, vote yea, Women's Health Protection Act of 2021...)
(Donald Trump, country of citizenship, United States of America)	(2020 United States presidential election in Colorado, successful candidate, Joe Biden)
(Donald Trump, owner of, Kingdom 5KR)	(Yemen, Host a visit, Barack Obama, 01/01/2010)

made to construct a politics-related knowledge graph, but it concentrates on the congressional employment status, neglecting the broader spectrum of behavior-related insights. Because of these limitations, we propose to construct a knowledge graph that is both **political domain-specific**, with a keen emphasis on political knowledge, and **multi-view**, covering diverse situations within U.S. politics. Generally, we start with extracting U.S. political conceptual knowledge in Wikidata KG [46] and extend it by incorporating factual knowledge, thereby ensuring a more comprehensive and nuanced understanding of political dynamics. Table 1 shows examples of conceptual knowledge and factual knowledge in MVPKG.

2.1 Political Conceptual Knowledge

To begin with, we select entities within U.S. politics, including the President and Cabinet members, Congressional legislators, Governors, and various government offices, as the seed entities. Subsequently, we manually retrieve their unique QID identifiers from Wikidata [46] and proceed to query all 1-hop facts to form a KG subset, which is named **baseKG**. This process ends with 71,646 entities, 368 relations and 103,174 triples.

2.2 Political Factual Knowledge

We further expand our knowledge graph with factual knowledge. Our objective is to ensure that the knowledge graph is multi-view, effectively covering key facets of U.S. politics, including legislation, elections, and diplomatic events spanning the past few decades. To achieve this, we have sourced data from a variety of resources and structured it for integration into the knowledge graph. Specifically, we use Legiscan API[1] to obtain legislative information including bills, sponsorship details and voting records of legislators. Data related to elections has been primarily collected from public sources such as Ballotpedia,[2] Wikipedia[3] and Cha, Kuriwaki, and Snyder [7]. This data contains election results for various offices including the President, Congressional House, Congressional Senate, Governor, State Houses, State Senate, and Mayors. For diplomatic events, records of interactions involving socio-political actors have been extracted, with a specific focus on those related to U.S. politics [5].

Most of the data crawled is structured or semi-structured. We organize the original data through certain rules so that all facts are expressed in the form of (subject, predicate, object) to conform to the storage format of the knowledge graph. Note that event data includes timestamps, which do not impact subsequent usage, as they can be treated as part of the factual context. As shown in Table 1, these historical factual facts provide clues for understanding political actors and events.

After acquiring subgraphs from each of these perspectives, we employ the entity linking tool [1, 17] to align entities with those already present in baseKG. Additionally, we further use BERT [22] to encode entities, merge entities with a similarity exceeding 0.95 to existing entities, and add the unmatched entities into baseKG. As a result, we obtain a comprehensive knowledge graph - **MVPKG**, composed of 116,176 entities, 602 relations and 1,857,410 triples.

[1]https://legiscan.com/
[2]https://ballotpedia.org/
[3]https://www.wikipedia.org/

3 EMPOWERING LARGE LANGUAGE MODELS WITH POLITICAL KNOWLEDGE

Figure 2 illustrates the overview of our proposed framework. We first acquire local evidence by entity-centric exploration and semantic-based filtering. Following this, we aggregate the local evidence to derive the global knowledge in a hidden space. To enhance interpretability and adapt to more LLMs, we also offer an explicit solution to express global knowledge in natural language. Finally, we perform prompt engineering to inject both local and global knowledge into LLMs' understanding and reasoning process.

3.1 Knowledge Acquisition

Given a question q, our goal is to retrieve a sub-graph $\mathcal{G}_q$ consisting of a set of fact triples $\{(\mathrm{e}_h, \mathrm{r}, \mathrm{e}_t)\}$ from an external KG $\mathcal{G} = \{(\mathrm{e}_h, \mathrm{r}, \mathrm{e}_t)\} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are sets of entities and relations, and e_h, r and e_t stand for the head entity, relation and tail entity, respectively.

3.1.1 Entity-centric Evidence Exploration. Since some key entities such as individuals and organizations are crucial in political scenarios, our initial step is to extract entities mentioned in the given question. Entity matching is implemented by existing entity-linking techniques [1, 17, 27]. This process yields an entity set $\mathcal{E}_q$ for exploration of evidence. We regard all the 1-hop fact triples associated with entities in $\mathcal{E}_q$ as the candidate fact triples, forming the candidate subgraph in Figure 2.

3.1.2 Semantic-based Evidence Filter. To simplify the process, one might consider injecting all candidate evidence related to entities directly into LLMs. However, this method suffers from limitations on input length and the potential for introducing noise, given the substantial number of associated triples, many of which might not be relevant to the question at hand. To address this challenge, we propose to further filter the evidence based on semantics. Firstly, we verbalize each fact triple which involves converting symbolic triples into text strings. We achieve this by concatenating the names of the head entity, relation and tail entity.

After verbalization, each fact triple can be regarded as a document and we can apply dense retrieval patterns [21, 54] to retrieve relevant evidence based on embedding similarities. To elaborate, we use the same encoder to embed both question and fact triples and compute their similarity. In this way, for each triple, we can define a retrieval score as the inner product between the embeddings of the given question q and the candidate triple t, as follows:

$$s(t \mid q) \propto \exp\left(d(t)^\top d(q)\right) \tag{1}$$

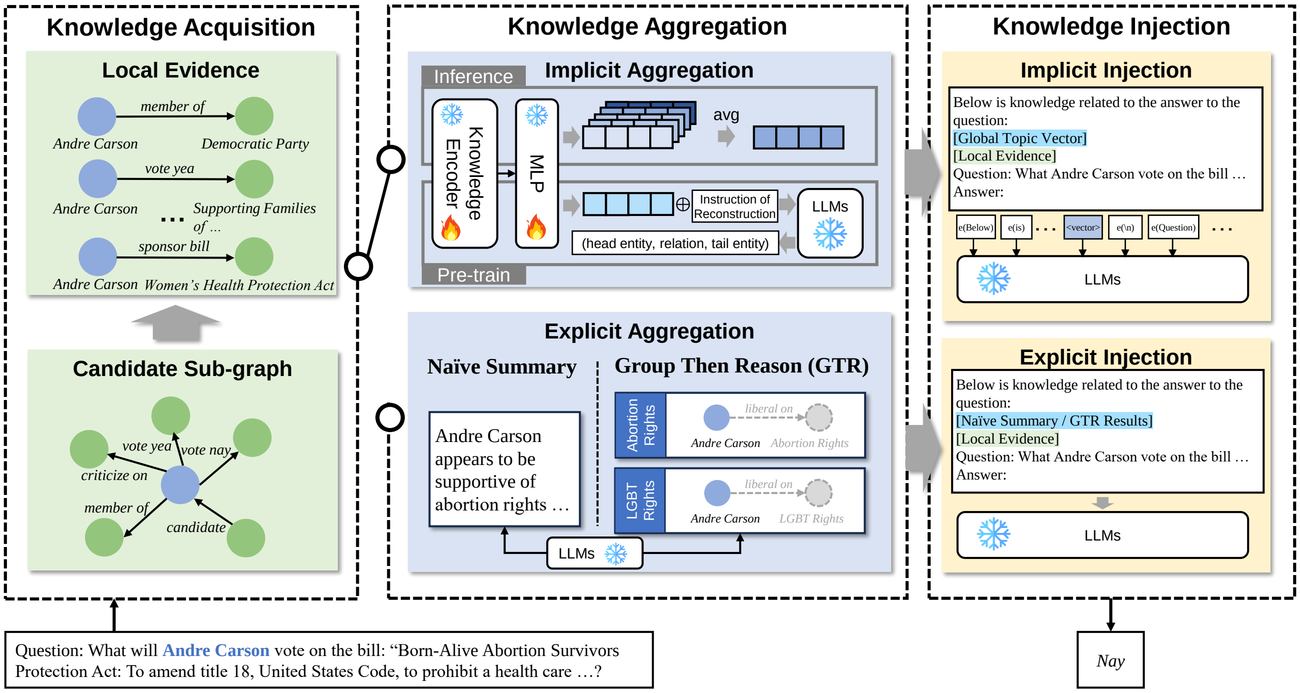

Figure 2: After constructing the multi-view political knowledge graph (MVPKG), we empower large language models with political knowledge through knowledge acquisition, aggregation, and injection. For knowledge aggregation, we implement two options: (1) implicit aggregation with embedding techniques, and (2) explicit aggregation with natural language.

where d is the embedding function. Subsequently, we only reserve top-K fact triples $\mathcal{G}_l = \{(e_h, r, e_t)\}$ as our local evidence, where K is a pre-defined hyper-parameter.

3.2 Knowledge Aggregation

After knowledge acquisition, most existing work [3, 43, 52] directly prompts LLMs with plain text or reformatted paths. However, they overlook the semantic relationships that underlie the facts within the knowledge graph. In this section, we describe how we provide LLMs with more comprehensive knowledge. This enhancement is achieved by aggregating local evidence to form global and more generalized knowledge compared to the fine-grained evidence, to better deal with the situations where direct facts are not readily matched, and relying only on local evidence proves insufficient.

3.2.1 Implicit Aggregation. Intuitively, we can aggregate the semantic information of retrieved facts in the embedding space. Here, we introduce another language model as a global knowledge encoder. Concretely, we use the knowledge encoder to encode the retrieved fact triples separately and average the sentence embedding as the global topic vector. This topic vector is further processed through a multilayer perceptron (MLP), to ensure alignment with the semantic space of LLMs, as follows:

$$\boldsymbol{v} = \frac{1}{K}\sum_{k=1}^{K} MLP(Encoder(t_k)) \tag{2}$$

where t_k is the k-th triple after evidence filtering in Section 3.1.2.

This topic vector $\boldsymbol{v}$ serves as a soft prompt to facilitate LLMs in answering the questions effectively. However, it is worth noting that a knowledge encoder and an MLP with random initialization or general pre-training may perform poorly in transforming information from the space of the original knowledge encoder to that of the target LLMs. This is due to a lack of specific training for this purpose. Thus, we propose a *fact reconstruction* task to pre-train the knowledge encoder and MLP components.

Specifically, we use the knowledge encoder to acquire the sentence embedding of each single fact triple and prompt LLMs to reconstruct the texts of the given triple based on the embedding. This task enables the knowledge encoder and MLP to express fact-related information in LLMs' space, without additional annotated data. In essence, we leverage signals from a fixed language model to train the encoder, using a language modeling objective:

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{v}) = \prod_{t=1}^{T} p\left(y_t \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{y}_{0:t-1}\right) \tag{3}$$

where $\boldsymbol{y} = [y_1, ..., y_T]$ is the output response, *i.e.*, the text of input triple, $\boldsymbol{x} = [x_1, ..., x_N]$ is the prompt instructing LLMs to reconstruct the triple based on the given vector $\boldsymbol{v}$. Note that this process can be applied to any set of triples, and once the knowledge encoder is trained, it can be adapted to the frozen LLMs for inference directly.

3.2.2 Explicit Aggregation. Although aggregating vectors in a hidden space is straightforward, explaining what knowledge the vectors actually represent is not trivial. Relying on the strong reasoning and generation ability of large language models, we aggregate the facts in an explicit manner, where the derived knowledge is expressed in natural language, to provide a global view of the local evidence. This method is more flexible since it can be applied to black-box LLMs like ChatGPT [38], where inputs in the form of vectors are often not acceptable. The simplest way to achieve this goal is to prompt LLMs to reason and summarize the given evidence in natural language $\boldsymbol{w} = [w_1, ..., w_i]$, similar to the principles behind many prior studies [23, 34, 51]. We achieve this by instructing LLMs with the question, *"What can you infer from the following facts?"* This variant is named **Naive Summary**.

While the idea is intuitive, the dispersion of evidence can result in ambiguous and cluttered summaries. Often, these summaries may merely consist of repetitions or abstracts of individual facts. To address this issue, we introduce a **Group Then Reason (GTR)** strategy. Specifically, we add the instruction to prompt LLMs to divide the evidence into several groups $g_1, ..., g_M$ according to the topical information, and summarize the reasoning result of each group into new fact triples $\mathcal{G}_h = \{(\mathrm{e}_h, \mathrm{r}, \mathrm{e}_t)\}$. Symbolically, this approach generates new pseudo-entities that convey topical information and their relationships with existing entities. This enhances the potential for answering questions that may not have direct matches in the knowledge graph. By organizing evidence into topical categories, it becomes easier for LLMs to reason about implicit knowledge hidden within the evidence, such as a politician's ideology or attitudes towards subjects within the same category.

3.3 Knowledge Injection

Once we have collected the local evidence and aggregated global view, the next step is to inject the knowledge, allowing LLMs to provide answers that are rooted in the associated external knowledge. For explicit aggregation, we integrate the verbalized local evidence $\mathcal{G}_l$ and the aggregated result $\boldsymbol{w}$ or $\mathcal{G}_h$ using a pre-defined instruction template. This prompt is then placed at the beginning of the input question $\boldsymbol{q}$, to stimulate LLMs to generate answers conditioned on the provided knowledge. The process can be formalized as $p(\boldsymbol{y} \mid [\mathcal{G}_h, \mathcal{G}_l, \boldsymbol{q}])$, where $[\cdot]$ denotes concatenation.

For implicit aggregation, inspired by prompt tuning [25] and P-tuning [32], we regard the global topic vector as a soft token and concatenate it with token embeddings derived from the verbalized local evidence $\mathcal{G}_l$ and the question. The resulting sequence of token embeddings is then fed into the transformer layers of the LLMs.

Note that we are following the zero-shot setting, where we do not possess any labeled samples or train models. This differs from supervised learning [2, 20], where models are trained with a set of (question, answer) pairs or (question, ground-truth facts) pairs.

4 EXPERIMENTS

To showcase that MVPKG and the proposed framework for knowledge integration can generally assist various tasks in the political domain, we conduct comprehensive experiments.

4.1 Experiment Settings

4.1.1 Tasks and Datasets. We employ three datasets representing various political scenarios for the assessment. **RCVP** [36] is a congressional roll-call vote prediction dataset. We further collect 7,927 voting records in 2023 from Legiscan for evaluation. The historical records are integrated into KGs to align with the time-based setting proposed in Mou et al. [36]. As per the conventions outlined in prior studies [36, 37], we report the macro F1 score for this binary classification task. **ICEWS** [5] contains political diplomatic events where we reserve 9,322 samples from 01/01/2023 to 04/10/2023 for evaluation. Following Zhu et al. [59], we formulate the task as predicting either the subject or the object of each event. For each question, we randomly sample three negative entities in the same category with the ground-truth answer to form a multiple-choice setting. Accuracy is reported. **StaId** represents a statement identification task curated in this paper. We sample 4,000 tweets from Mou et al. [37] and create questions that revolve around determining whether a given statement on a specific issue was posted by a particular politician. This task assesses the capabilities of LLMs to comprehend politicians' attitudes on various issues. Macro F1 is reported for this binary classification task.

4.1.2 Compared Methods.

Baselines without External Knowledge.

- **Vanilla**, *i.e.*, providing questions directly to LLMs such as Llama2 [45], Vicuna [58] and GPT-3.5-turbo [38], without the integration of any knowledge.
- **GKP** [29] extracts knowledge from LLMs themselves and then prompts LLMs with the generated knowledge.
- **RECITE** [44] recites relevant passages from LLMs' own memory, and then produces the final answers.

Baselines with Local Evidence Only.

- **KAPING** [3] retrieves the knowledge based on similarity and prompts the textual triples to LLMs.
- **MindMap**$_{route}$ [52] clusters the retrieved triples into structured pathways like *2020 U.S. state House of Representatives elections in District 75 of Iowa->candidate->(Ruby Bodeker, Thomas Gerhold)*, which is subsequently prompted to LLMs.
- **MindMap**$_{lang}$ [52] prompts LLMs to describe the evidence route in natural language and leverage the generated content for further prompting.
- **MindMap** [52] prompts LLMs to answer the questions and meanwhile describe the evidence route and construct a decision tree-like mindmap.

Our Methods.

- **PEG**$_{imp}$, our method with implicit aggregation.
- **PEG**$_{exp\_sum}$, our method with explicit aggregation, where the global knowledge is generated through Naive Summary.
- **PEG**$_{exp\_GTR}$, our method with explicit aggregation, where the global knowledge is generated through GTR.

4.1.3 Implementation Details. We use several LLMs to verify the effectiveness of our framework, including Llama2-7b-chat [45], Vicuna-7b-v1.1 [58] and GPT-3.5-turbo [38]. We use ReFinED [1] for entity linking and a document-pretrained distillbert[4] as the retriever for semantic filtering. We use the KAPING [20] methods to arrange local evidence. Top-10 facts are reserved for knowledge graph integration. When evaluating white-box LLMs including Llama2 and Vicuna, we follow Li et al. [28] to concatenate each

[4]https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v3

Table 2: Main results of white-box large language models. The best scores are emphasized in bold.

Methods	RCVP		ICEWS		StaId	
	Llama2	Vicuna	Llama2	Vicuna	Llama2	Vicuna
Vanilla	40.07	37.17	23.98	22.88	57.10	49.57
GKP [29]	42.95	35.71	29.40	24.47	56.56	52.73
RECITE [44]	36.94	36.15	28.08	24.48	56.01	**56.93**
KAPING [3]	44.66	42.92	39.80	36.06	53.67	53.84
MindMap$_{route}$ [52]	43.41	43.73	40.07	36.12	52.67	53.85
MindMap$_{lang}$ [52]	44.67	42.44	37.87	33.94	53.03	45.84
MindMap [52]	43.45	40.24	33.38	29.95	55.42	55.27
PEG$_{imp}$	**52.56**	**44.56**	37.47	34.25	56.36	52.07
PEG$_{exp_sum}$	47.77	42.75	38.60	35.06	**58.67**	53.67
PEG$_{exp_GTR}$	47.49	44.16	**40.10**	**36.51**	55.11	54.76

Table 3: Main results of GPT-3.5. The best scores are emphasized in bold. Note that due to the constraint of the input format, we do not include results of PEG$_{imp}$.

Methods	RCVP	ICEWS	StaId
Vanilla	34.11	19.40	33.24
GKP [29]	20.43	15.40	45.32
RECITE [44]	15.79	21.40	35.57
KAPING [3]	37.83	20.80	36.99
MindMap$_{route}$ [52]	32.60	28.00	35.03
MindMap$_{lang}$ [52]	38.57	28.40	42.19
MindMap [52]	38.72	25.60	23.38
PEG$_{exp_sum}$	40.62	26.40	42.12
PEG$_{exp_GTR}$	**41.21**	**28.60**	**46.21**

candidate answer with the input and compare the language modeling likelihood to determine the answer for a stable evaluation. When it comes to black-box LLMs such as GPT-3.5, we evaluate based on the generated results since likelihood is not available. Unless otherwise specified, all KG-enhanced methods use knowledge sourced from MVPKG. More details can be found in Appendix A.1.

4.2 Experiment Results

4.2.1 Main Results. Table 2 presents the primary results across various tasks and white-box language models. In general, our proposed methods *consistently* outperform other baselines. It is important to note that the generated knowledge model (GKP) is not significantly superior to the vanilla knowledge-free model, since in most cases of our political tasks, LLMs can not generate relevant and accurate knowledge about future facts, limiting their assistance in improving answers. Conversely, KG-augmented methods clearly outperform the naive models, underscoring the value of the knowledge contained within MVPKG for addressing political tasks. Among these methods, our methods demonstrate distinct advantages, suggesting that simply arranging local facts or describing the structural relations among facts is **insufficient** for tackling complex tasks like vote prediction. Integrating global knowledge can lead to **substantial enhancements**. Additionally, building a mindmap proves to be a challenging endeavor, especially for smaller-scale LLMs. Consequently, MindMap falls short of achieving superior results compared

Table 4: Results of adding global knowledge to different patterns of local evidence.

Evidence Format	Integration Method	RCVP	ICEWS	StaId
KAPING	w/o global knowledge	44.66	39.80	53.67
	PEG$_{imp}$	**52.56**	37.47	56.36
	PEG$_{exp_sum}$	47.77	38.60	**58.67**
	PEG$_{exp_GTR}$	47.49	**40.10**	55.11
Mindmap$_{route}$	w/o global knowledge	43.41	40.07	52.67
	PEG$_{imp}$	**48.67**	39.84	**55.63**
	PEG$_{exp_sum}$	47.48	40.31	53.49
	PEG$_{exp_GTR}$	48.11	**40.37**	53.01
Mindmap$_{lang}$	w/o global knowledge	44.67	37.87	53.03
	PEG$_{imp}$	45.03	38.17	53.90
	PEG$_{exp_sum}$	45.46	**39.68**	**54.52**
	PEG$_{exp_GTR}$	**45.65**	39.63	49.29

to its simpler counterparts, MindMap$_{route}$ and MindMap$_{lang}$. In contrast, summarizing and reasoning about the local evidence is not only more feasible but also cost-effective.

Through comparison, it is evident that various datasets and tasks necessitate distinct aggregation strategies. In the case of RCVP, **PEG** with implicit aggregation proves most effective, primarily because the retrieved facts predominantly consist of historical voting records with lengthy text, and the explicitly summarized content may include irrelevant information. Conversely, for ICEWS and StaId, **PEG** with explicit aggregation exhibits a slight advantage. This difference could be attributed to the retrieved facts being less concentrated compared to the voting records in RCVP, making the averaging of facts in the latent space potentially less meaningful.

Table 3 provides the results of GPT-3.5. Relying on the rich internal knowledge of GPT-3.5, the advantage of the GKP method becomes evident in the StaId dataset. However, it still struggles to handle tasks like RCVP. Comparatively, our methods show *consistent* superiority in assisting this black-box LLM across diverse tasks. The improvement brought by PEG$_{exp_GTR}$ is more pronounced for GPT-3.5 than for Llama2 and Vicuna, as GPT-3.5 poses stronger capabilities to group and summarize local information.

4.2.2 Effectiveness of MVPKG. To demonstrate the effectiveness of our MVPKG, we compare it with other generic knowledge graphs including Wikidata KG [46], *i.e.*, baseKG in this paper, and YAGO [40] and political knowledge graph KGAP proposed in Feng et al. [11]. Figure 3 shows the performance paired with different KGs on Llama2. The results suggest that domain knowledge graphs including KGAP and our MVPKG are more practical than generic KGs in understanding the political actors and events since they assist models in learning from related factual history. Moreover, our knowledge graph is more comprehensive than KGAP, encompassing various aspects of U.S. politics. Consequently, it exhibits strengths across diverse datasets. Surprisingly, Wikidata demonstrates competitive performance on the StaId dataset. This could be attributed to the fact that some statements' content is partially linked to fundamental attributes of politicians, such as party affiliation and home state.

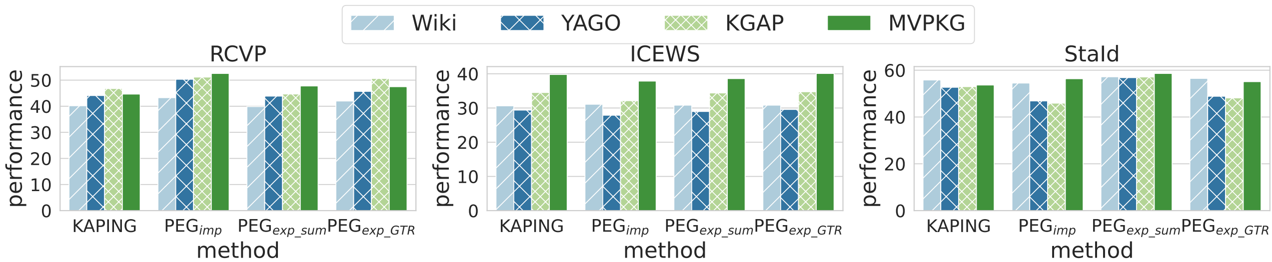

Figure 3: Performance of Llama2 when using different knowledge graphs for knowledge integration.

4.2.3 Effectiveness of Global Knowledge. In order to illustrate the effectiveness of global knowledge, we test our **PEG** framework with different patterns of local evidence, *i.e.*, KAPING, MindMap$_{route}$ and MindMap$_{lang}$. Table 4 shows the results of this ablation study. Overall, global knowledge works across three different forms of local evidence. PEG$_{exp_sum}$ exhibits the most consistent performance, since the global knowledge in a format of natural languages is the most straightforward for a language model to comprehend. While for PEG$_{exp_GTR}$ on Mindmap$_{route}$ and Mindmap$_{lang}$, LLMs need to understand at least two forms of knowledge, *i.e.*, local evidence in forms of path or language and global knowledge in forms of triple. Thus, the gain is unstable. In contrast, PEG$_{exp_GTR}$ with KAPING clearly improves the KAPING baseline, since both local and global knowledge is expressed in fact triples.

In response to the challenge of limited domain knowledge comprehension, some researchers [9, 56] have turned to instruction tuning [33] using domain-specific data. We compare our methods with this line of approaches that fine-tune LLMs to solve domain tasks. Results in Appendix A.5 reveal that instruction tuning does not exhibit significant advantages over our external knowledge integration approach. Moreover, the augmentation of LLMs with knowledge graphs, instead of resorting to additional training, proves to be a more flexible and cost-effective solution, particularly in addressing the rapidly evolving political landscape.

5 FURTHER ANALYSIS

In this section, we conduct more experiments to implement an in-depth analysis of the PEG framework and global knowledge.

5.1 Explainability of Global Knowledge

Compared to explicit methods where reasoning results are expressed in natural language, the implicit method is much more difficult to explain. Intuitively, the aggregated vector contains information on one or more aspects related to the central entity with respect to the given question. To validate this, we take the RCVP dataset as an example for analysis by visualizing the PCA-transformed representation of the global knowledge vector. Figure 4 depicts the global knowledge vector after dimensionality reduction of questions asking to predict different Congressional members' votes on two bills respectively. We label each member's position on abortion issues (*Pro-choice* and *Pro-life* in Figure 4) and transgender issues (*For Transgender Rights* and *Not For Transgender Rights* in Figure 4) based on information provided by public websites.[5] [6] The visualization shows that samples with similar positions cluster together. This indicates that the global knowledge vector aggregated by the knowledge encoder, using factual traces, can reflect the attitudes of key individuals on specific topics or issues. With this information, models can more easily deduce how members are likely to vote on emerging bills related to the same issue.

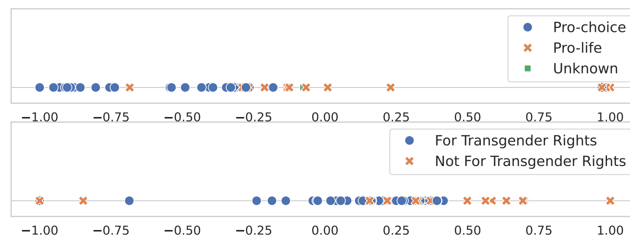

Figure 4: The PCA outputs of the global knowledge vectors corresponding to questions asking to predict Congressional members' votes on the Born-Alive Abortion Survivors Protection Act (top) and the Protection of Women and Girls in Sports Act of 2023 (bottom).

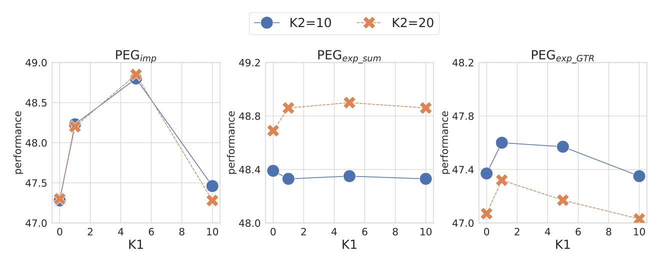

Figure 5: Performance with varying amounts of knowledge, where we change the number of fact triples K1 for local evidence and K2 for global knowledge.

5.2 Impact of the Amount of Knowledge

To explore the influence of the information load on both local and global knowledge, we vary the number of facts used to form local evidence and global knowledge. Figure 5 displays the average results

[5] https://justfacts.votesmart.org/
[6] https://www.ontheissues.org/

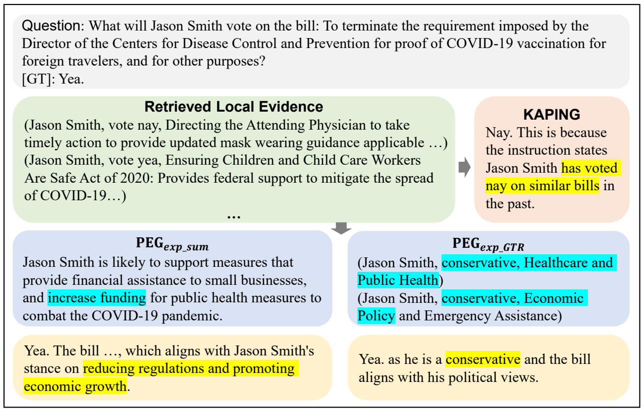

Figure 6: A case from the RCVP dataset. Clues utilized by LLM inference and reasons for answers are highlighted.

of three datasets when using top $K1$ facts for local evidence and top $K2$ facts for global knowledge through different aggregation methods on LLAMA2. Firstly, the performance shows an increasing trend followed by a decrease as the quantity of local evidence changes in most cases. This trend occurs because including more clues can lead to improvement initially, but as the amount of provided facts increases, LLMs become distracted by irrelevant fact triples. This phenomenon is not significant for PEG$_{exp_sum}$, possibly because it extracts more relevant information during the summary process. Also, when relying solely on global knowledge (i.e., $K1 = 0$), these methods, particularly the explicit ones, can still produce competitive results. In contrast to general KGQA tasks, where the ground-truth answer triple is often expected in the retrieved results, our approach leverages both local and global knowledge to provide information rather than searching for specific answer triples. Furthermore, when $K2$ increases, the methods exhibit varying trends. PEG$_{exp_sum}$ demonstrates a performance improvement, whereas PEG$_{exp_GTR}$ experiences a decline, mainly due to the drop in GTR performance when more facts with longer texts are introduced. This indicates that the effectiveness of knowledge aggregation methods can be influenced by the quantity of knowledge used.

5.3 Case Study

Figure 6 shows a case from the RCVP dataset, where we further ask LLM why it outputs the answer. Even provided with relevant information, KAPING [3] cannot answer correctly. This is because bills of similar topics might express different leanings, but LLM does not fully comprehend their relationships. This again highlights the challenges in reasoning from such local evidence. In contrast, PEG$_{exp_}$ and PEG$_{exp_GTR}$ provide more direct information.

6 RELATED WORK

Knowledge-augmented LLMs. To mitigate the hallucination of LLMs, recent works have leveraged external knowledge for LLM inference. Works represented by REALM [16], RAG [26] and Replug [42] propose to retrieve documents and augment LLMs with the unstructured corpora. Compared to documents, knowledge graphs are less constrained by the limited input length of LLMs and can express more explicit knowledge in a more compact form of fact triples. Previous research on KG-augmented language models focuses on designing additional modules and training objectives to incorporate knowledge in different stages of language models [31, 39, 48–50, 57]. With the recent progress of LLMs, the research paradigm is shifting to prompting fixed LLMs without additional training. Baek et al. first retrieve triples from knowledge graphs based on similarity and prompt the triples to LLMs to handle question answering. Guo et al. [15] and Wang et al. [47] explore prompting LLMs for graph mining. To fill the gap in understanding structural information, Sun et al. [43] iteratively retrieve triples and construct reasoning paths and Wen et al. [52] prompt LLMs to generate a decision tree-like mindmap to visualize the reasoning process. Although reporting positive results, existing methods focus on directly stacking all facts or paths between facts, ignoring the semantic relationships between facts, *i.e.*, the global thematic information. Also, these methods are primarily designed and tested on general or medical domain QA datasets, where the answers are typically entities that can be directly matched within the knowledge graphs. Consequently, plain text answers or reasoning paths can provide effective solutions. However, when applied to more challenging tasks such as prediction, they may fall short since the direct facts or paths needed for accurate answers can no longer be readily found. We aim to address this issue by experimenting with a simple yet effective approach - aggregating local facts to generate higher-level global knowledge that is both more directly relevant to the questions and capable of providing better signals for LLMs.

Political Actor Modeling and Opinion Mining. Modeling political actors and understanding their discourse is at the core of computational political science. Early research focused on statistical analysis of roll call data to estimate the ideology of political actors. One of the most widely used methods for vote-based analysis is the Ideal Point Model [8], which reveals how the divisions among legislators reflect their partisan affiliations. Researchers have expanded upon this model by incorporating the bill texts to enhance performance [13, 24]. Recently, more abundant contextual information such as co-sponsorship network [55], hashtag network [36] and relations of contributors [10] have been introduced. Although these metadata have proven effective, collecting them in large quantities is expensive due to the complex and diverse data formats. Mou et al. [37] proposed a unified scheme by injecting social information in the pre-training stage and using languages only to represent political actors to solve downstream tasks. Considering LLMs' strong ability in understanding and reasoning, we do not train the models but construct a multi-view knowledge graph where political knowledge is expressed in a unified format of triple, whose source records are publicly available and continuously updated.

7 CONCLUSION

In this study, we construct a political knowledge graph covering diverse facets of U.S. politics, and we introduce the **PEG** framework to address tasks in political actor modeling and opinion mining. We unify the local and global knowledge to alleviate the issue when direct answers cannot be found in constructed knowledge graphs. Experiments demonstrate the effectiveness of our method.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (No. 6217020551) and Science and Technology Commission of Shanghai Municipality Grant (No.21DZ1201402).

REFERENCES

[1] Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. Refined: An efficient zero-shot-capable approach to end-to-end entity linking. *arXiv preprint arXiv:2207.04108* (2022).

[2] Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. 2023. Direct Fact Retrieval from Knowledge Graphs without Entity Linking. *arXiv preprint arXiv:2305.12416* (2023).

[3] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. *arXiv preprint arXiv:2306.04136* (2023).

[4] 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*. 1247–1250.

[5] Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS Coded Event Data. https://doi.org/10.7910/DVN/28075

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Jeremiah Cha, Shiro Kuriwaki, and James M. Jr. Snyder. 2021. Candidates in American General Elections. https://doi.org/10.7910/DVN/DGDRDT

[8] Joshua Clinton, Simon Jackman, and Douglas Rivers. 2004. The statistical analysis of roll call data. *American Political Science Review* 98, 2 (2004), 355–370.

[9] Yongfu Dai, Duanyu Feng, Jimin Huang, Haochen Jia, Qianqian Xie, Yifang Zhang, Weiguang Han, Wei Tian, and Hao Wang. 2023. LAiW: A Chinese Legal Large Language Models Benchmark (A Technical Report). *arXiv preprint arXiv:2310.05620* (2023).

[10] Maryam Davoodi, Eric Waltenburg, and Dan Goldwasser. 2020. Understanding the language of political agreement and disagreement in legislative texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5358–5368.

[11] Shangbin Feng, Zilong Chen, Wenqian Zhang, Qingyao Li, Qinghua Zheng, Xiaojun Chang, and Minnan Luo. 2021. Kgap: Knowledge graph augmented political perspective detection in news media. *arXiv preprint arXiv:2108.03861* (2021).

[12] Matthew Gentzkow, Jesse M Shapiro, and Matt Taddy. 2019. Measuring group differences in high-dimensional choices: method and application to congressional speech. *Econometrica* 87, 4 (2019), 1307–1340.

[13] Sean M Gerrish and David M Blei. 2011. Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*.

[14] Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. Rumoureval 2019: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1809.06683* (2018).

[15] Jiayan Guo, Lun Du, and Hengyu Liu. 2023. GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking. *arXiv preprint arXiv:2305.15066* (2023).

[16] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[17] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[19] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299* (2022).

[20] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation. *arXiv preprint arXiv:2305.18846* (2023).

[21] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[22] 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*. 4171–4186.

[23] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[24] Peter Kraft, Hirsh Jain, and Alexander M Rush. 2016. An embedding model for predicting roll-call votes. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 2066–2070.

[25] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[27] Belinda Z Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. *arXiv preprint arXiv:2010.02413* (2020).

[28] Zejun Li, Ye Wang, Mengfei Du, Qingwen Liu, Binhao Wu, Jiwen Zhang, Chengxing Zhou, Zhihao Fan, Jie Fu, Jingjing Chen, et al. 2023. ReForm-Eval: Evaluating Large Vision Language Models via Unified Re-Formulation of Task-Oriented Benchmarks. *arXiv preprint arXiv:2310.02569* (2023).

[29] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2021. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387* (2021).

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[31] 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*, Vol. 34. 2901–2908.

[32] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. GPT understands, too. *AI Open* (2023).

[33] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688* (2023).

[34] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. LLM-Rec: Personalized Recommendation via Prompting Large Language Models. *arXiv preprint arXiv:2307.15780* (2023).

[35] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*. 31–41.

[36] Xinyi Mou, Zhongyu Wei, Lei Chen, Shangyi Ning, Yancheng He, Changjian Jiang, and Xuan-Jing Huang. 2021. Align voting behavior with public statements for legislator representation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1236–1246.

[37] Xinyi Mou, Zhongyu Wei, Qi Zhang, and Xuan-Jing Huang. 2023. UPPAM: A Unified Pre-training Architecture for Political Actor Modeling based on Language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 11996–12012.

[38] OpenAI. 2023. ChatGPT (Mar 14 version). https://chat.openai.com/chat

[39] 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* (2023).

[40] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*. Springer, 583–596.

[41] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156* (2018).

[42] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652* (2023).

[43] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model with Knowledge Graph. *arXiv preprint arXiv:2307.07697* (2023).

[44] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-Augmented Language Models. In *The Eleventh International Conference on Learning Representations*.

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[46] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[47] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can Language Models Solve Graph Problems in Natural Language? *arXiv preprint arXiv:2305.10037* (2023).

[48] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuan-Jing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 1405–1418.

[49] Siyuan Wang, Zhongyu Wei, Jiarong Xu, Taishan Li, and Zhihao Fan. 2023. Unifying structure reasoning and language model pre-training for complex reasoning. *arXiv preprint arXiv:2301.08913* (2023).

[50] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics* 9 (2021), 176–194.

[51] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[52] Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729* (2023).

[53] Zhiping Xiao, Weiping Song, Haoyan Xu, Zhicheng Ren, and Yizhou Sun. 2020. TIMME: Twitter ideology-detection via multi-task multi-relational embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2258–2268.

[54] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[55] Yuqiao Yang, Xiaoqiang Lin, Geng Lin, Zengfeng Huang, Changjian Jiang, and Zhongyu Wei. 2021. Joint representation learning of legislator and legislation for roll call prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 1424–1430.

[56] Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023. HuatuoGPT, towards Taming Language Model to Be a Doctor. *arXiv preprint arXiv:2305.15075* (2023).

[57] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129* (2019).

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685 [cs.CL]

[59] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4732–4740.

Table 6: Running Efficiency Analysis. RT local refers to the relative time of running models locally than that of Vanilla Llama2-7B and RT API refers to the relative time of running models using the gpt-3.5-turbo interface to that of Vanilla GPT-3.5-turbo. We set the max_new_token for the generation of MINDMAP to twice that of other methods because the mindmap costs numerous tokens.

Methods	RT local	RT API	Process
Vanilla	1.00	1.00	Final Inference
GKP	10.84	2.10	Intermediate Generation+Final Inference
Recite	13.05	3.11	Intermediate Generation+Final Inference
KAPING	1.68	1.88	Retrieval+Final Inference
MindMap$_{path}$	5.21	1.95	Retrieval+Final Inference
MindMap$_{lang}$	13.89	4.99	Retrieval+Intermediate Generation+Final Inference
MindMap	24.15	13.86	Retrieval+Intermediate Generation+Final Inference
PEG$_{imp}$	8.99	-	Retrieval+External Encoding+Final Inference
PEG$_{exp_sum}$	13.78	2.52	Retrieval+Intermediate Generation+Final Inference
PEG$_{exp_GTR}$	13.57	3.42	Retrieval+Intermediate Generation+Final Inference

A EXPERIMENT DETAILS

A.1 Hyperparameters

Table 5: Hyperparameter settings.

Hyperparameter	**Value**
Knowledge Encoder Training	
batch size	1,024
epochs	5
learning rate for encoder	2e-5
learning rate for MLP	2e-4
warmup ratio	0.1
encoder	distillbert [7]
Explicit Aggregation	
max tokens for summary generation	128
max tokens for GTR generation	128
do sample	False
temperature	1
num beams	1
LLM inference	
K	10
VectorDB for retrieval	FAISS

We present the hyperparameter settings in Table 5.

A.2 Evaluation

For evaluation on white-box LLMs, *i.e.*, Llama2 and Vicuna, we follow Li et al. [28] to implement a likelihood evaluation. Given knowledge context k, question q and options $C = \left\{c^i\right\}_{i=1}^{N}$, the answer prediction can be determined by the generation likelihood predicted by the evaluated model:

$$\hat{c} = \arg\max_{c^i \in C} P_\theta\left(c^i \mid k, q\right) \tag{4}$$

where P_θ is parameterized by the causal-LLM.

For evaluation on black-box LLMs, *i.e.*, GPT-3.5, we provide options in prompt and ask the model to output its choice, since likelihood is not applicable in the API. And we further write regular expressions to match answers to deal with situations when options are not explicitly output.

A.3 Running Efficiency Analysis

We conduct experiments on RCVP to analyze the running efficiency. The results in Table 6 show that the intermediate generation brings the main efficiency reduction, but this is unavoidable if we want to interact with LLMs. Also, the time spent on intermediate generation in our method is comparable to most generation-based methods (GKP, RECITE, MindMap), so we think it is acceptable. This can be partially alleviated by using OpenAI API-based LLMs or our proposed implicit (PEG_*imp*) variant. In the future, we will continue to explore how to interact with LLMs more efficiently.

A.4 Prompts

We illustrate prompt examples of PEG$_{exp_sum}$ and PEG$_{exp_GTR}$.

Prompt of PEG$_{exp_sum}$

What can you infer from the following facts?
Facts: {facts}
Inference:

Prompt of PEG$_{exp_GTR}$

Here are some fact triples in the form of (subject, predicate, object). Group the facts based on the topical information and summarize what you can infer from each group of facts into triples. Output the triples only.
Here is an example.
Input: {example_input}
Output: {example_output}

Try to output:
Input: {local evidence}
Output:

Table 7: Performance on vanilla Vicuna, instruction-tuned Vicuna and KG-enhanced Vicuna (best PEG variant is reported).

Method	**RCVP**	**ICEWS**	**StaId**
Vanilla	37.17	22.88	49.57
FT	38.81	23.89	39.18
PEG	**44.56**	**36.51**	**54.76**

A.5 Comparison with Domain Instruction-tuned LLMs

Facing the challenge of insufficient comprehension of domain knowledge, some researchers [9, 56] have employed instruction tuning [33] on domain-specific data. To compare directly providing external knowledge with internalizing knowledge through finetuning in the political domain, we curate an instruction dataset consisting of 28,187 samples, from 28 publicly available datasets across 11 tasks, such as stance detection [14, 35] and ideology detection [12, 53]. Subsequently, we use LoRA [18] to finetune Vicuna. As shown in Table 7, instruction tuning (FT) does not show obvious advantages over the external knowledge integration solution. Meanwhile, enhancing LLMs with KG instead of additional training appears to be a more adaptable and cost-effective solution for the rapidly evolving political landscape.