In-Context Learning with Topological Information for LLM-Based Knowledge Graph Completion

Anonymous Authors[1]

Abstract

Knowledge graphs (KGs) are crucial for representing and reasoning over structured information, supporting a wide range of applications such as information retrieval, question answering, and decision-making. However, their effectiveness is often hindered by incompleteness, limiting their potential for real-world impact. While knowledge graph completion (KGC) has been extensively studied in the literature, recent advances in generative AI models, particularly large language models (LLMs), have introduced new opportunities for innovation. In-context learning has recently emerged as a promising approach for leveraging pretrained knowledge of LLMs across a range of natural language processing tasks and has been widely adopted in both academia and industry. However, how to utilize in-context learning for effective KGC remains relatively underexplored. We develop a novel method that incorporates topological information through in-context learning to enhance KGC performance. By integrating ontological knowledge and graph structure into the context of LLMs, our approach achieves strong performance in the transductive setting i.e., nodes in the test graph dataset are present in the training graph dataset. Furthermore, we apply our approach to KGC in the more challenging inductive setting, i.e., nodes in the training graph dataset and test graph dataset are disjoint, leveraging the ontology to infer useful information about missing nodes which serve as contextual cues for the LLM during inference. Our method demonstrates superior performance compared to baselines on the ILPC-small and ILPC-large datasets.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SPIGM workshop at ICML 2024. Do not distribute.

1. Introduction

Knowledge graphs have emerged as a powerful framework for representing and reasoning over large-scale structured knowledge, with applications spanning question answering (Song et al., 2023; Yani & Krisnadhi, 2021), reccomendation systems (Chicaiza & Valdiviezo-Diaz, 2021) and decision making (Hogan et al., 2021). The effectiveness of knowledge graphs in these domains relies heavily on their completeness and accuracy. However, constructing and maintaining comprehensive knowledge graphs is a challenging task, often requiring extensive manual effort and domain expertise (Yan et al., 2016). To address this challenge, researchers have explored automated methods for Knowledge Graph Completion (KGC) (Shen et al., 2022; Chen et al., 2020). These methods leverage information extracted from various sources, such as text corpora (Wang et al., 2021; Shi & Weninger, 2018), web pages (Dong et al., 2014; Mitchell et al., 2018), and databases (Zou et al., 2014). These methods typically employ natural language processing techniques, such as named entity recognition and relation extraction (Li et al., 2022; Pawar et al., 2017), to identify new facts, integrate them into the existing knowledge graphs and identifying new relations between nodes that exist in the graph. The rapid growth of unstructured data has introduced new challenges in automated graph extension methods.

Recent advances in generative AI, particularly the emergence of Large Language Models (LLMs) have opened up new possibilities for extending knowledge graphs (Pan et al., 2024). Modern LLMs demonstrate impressive capabilities in understanding large texts (Gemini Team, 2024), generating natural language as well as reasoning (OpenAI, 2024) over complex information. These models are trained on vast amounts of unstructured data, allowing them to capture rich semantic knowledge and develop a deep understanding of various domains. Pretrained knowledge of LLMs and their reasoning capabilities can be harnessed to infer connections between nodes and relations predicting missing information in graphs. links in the graph, making LLMs well suited for task of KGC.

In this paper, we propose a novel approach that leverages graph topological information through in-context learning to enhance the KGC performance of LLMs. Our method is based on the intuition that the extensive knowledge embedded in pretrained language models can provide valuable insights and predictions for missing information in knowledge graphs. We introduce a two-step process: first, we construct an ontology from the knowledge graph using the LLM's domain understanding, capturing the types of nodes and relationships in the graph. By combining this ontological structure with the graph's topology and employing chain-of-thought (CoT) reasoning, we provide the LLM with context to make more informed predictions. Secondly, our algorithm leverages structured information from the graph, utilizing overlapping nodes between the missing knowledge triplets and the existing graph triplets, combined with the ontology, to generate candidate solutions for the missing information. Additionally, we consider alternative paths between the existing nodes and potential candidate nodes, thereby exploiting the complex topological structure of the graph. This comprehensive use of the graph's topological structure and the LLM's predictive capabilities leads to our method performing significantly better than the state-of-the-art baseline approaches.

Our contributions are as follows: (1) We propose a generative ontology creation method using LLMs to derive ontologies from raw knowledge graph data, capturing the types of nodes and relationships in the graph. (2) We leverage the generated ontology and the graph's topological information, including paths between nodes, to enhance link prediction. (3) By utilizing the ontology to identify candidate solutions for missing triplets and employing the LLM to select the correct solution, we improve KGC performance in both transductive and inductive settings. Importantly, our method requires no additional training, highlighting its efficiency and immediate applicability.

2. Related work

There exists a substantial body of work at the intersection of LLMs and knowledge graphs, extensively reviewed in (Pan et al., 2024). This section provides a detailed exploration of relevant literature.

Knowledge graph completion datasets Some of the popular knowledge graph completion datasets include Freebase, a comprehensive knowledge base integrated into Google's Knowledge Graph, from which FB15k (Bordes et al., 2013) and FB15k-237 (Toutanova et al., 2015) datasets are derived. WN18RR (Dettmers et al., 2018) is a subset of WordNet widely used for link prediction models. ILPC (Inductive Link Prediction Challenge) datasets (Galkin et al., 2022a) are also significant, featuring both small and large datasets designed for inductive reasoning tasks. Other domain-specific datasets include MEDCIN (med, 2024) covering biomedical entities and GeoNames (geo) focusing on geographical entities. These datasets collectively serve as crucial benchmarks for evaluating various aspects of knowledge graph completion models.

Link prediction in knowledge graphs One group of methods for link prediction in knowledge graphs involves the use of vector embeddings. (Kazemi & Poole, 2018) makes use of background knowledge when creating the embeddings. (Zhang et al., 2021) embeds head and tail entities into time and frequency domain spaces respectfully. (Zhang et al., 2020) shows an embedding scheme based on entity type hierarchies. Another family of methods is the use of trained deep neural network models. (Zhang & Chen, 2018), (Nguyen et al., 2022) and (Mohamed et al., 2023) demonstrate the use of GNNs for link prediction in graphs and knowledge graphs. (Neelakantan et al., 2015) uses RNNs for graph completion. Other methods focus on 'lifelong-learning' and the continual updating of knowledge graphs based on new information (Mazumder et al., 2019).

LMs for link prediction ans ontology creation Approaches such as BERTRL (Zha et al., 2022) and KGT5 (Saxena et al., 2022) treat each triplet in the knowledge graph as a textual sequence, refining models based on these sequences. These methods leverage information from language model parameters but do not explicitly integrate information extracted from knowledge graph during link prediction. In contrast, frameworks like Better Together (Chepurova et al., 2023) and KGT5-context (Kochsiek et al., 2023) incorporate node neighborhoods directly within the context of generative language models. Additionally, recent studies have explored the capability of large language models (LLMs) to create coherent ontologies. LLMs4OL (Giglou et al., 2023) systematically evaluated various LLMs, demonstrating that models fine-tuned for specific tasks consistently outperformed zero-shot methods. In another approach, Kommineni et al. (Kommineni et al., 2024) used LLMs to generate "competency questions," which were employed to develop ontologies for knowledge graphs.

3. Problem formulation

In this section, we introduce mathematical notations and formally define the problem of KGC using LLMs incoporating topological information.

3.1. Knowledge graph with ontology

Let $\mathcal{O}$ be an ontology and let $\mathcal{G}$ be the corresponding knowledge graph.

Ontology $\mathcal{O}$ can be defined as $\mathcal{O} = (\mathcal{C}, \mathcal{R}, \mathcal{E})$, where:

- $\mathcal{C}$ consists of ontology nodes, node categories of the nodes in the graph,

- $\mathcal{R}$ is the set of relations,
- $\mathcal{E}$ consists of unique triplets (c_i, r, c_j) where $c_i, c_j \in C$ and $r \in \mathcal{R}$.

Graph $\mathcal{G}$ can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{T})$, where:

- $\mathcal{V}$ is the set of nodes, where each node $v_i \in \mathcal{V}$ is associated with at least one category $c_{v_i} \in \mathcal{C}$.
- $\mathcal{R}$ is the set of relations,
- $\mathcal{T}$ consists of triplets formed according to the ontology triplets $\mathcal{E}$. For nodes v_i of category c_{v_i} and v_j of category c_{v_j}, $(v_i, r, v_j) \in \mathcal{T}$ such that $(c_{v_i}, r, c_{v_j}) \in \mathcal{E}$.

3.2. Knowledge graph completion

Knowledge Graph Completion (KGC) is the task of inferring missing information in a knowledge graph. Given a training knowledge graph $\mathcal{G}_{\text{train}} = (\mathcal{V}, \mathcal{R}, \mathcal{T}_{\text{train}})$ with some missing triplets $(v_i, r, v_j) \in \mathcal{T}_{\text{inference}}$, the objectives are to: (1) predict the missing relation $r \in \mathcal{R}$ between two existing entities $v_i, v_j \in \mathcal{V}$, i.e., $(v_i, ?, v_j)$; (2) predict the missing tail entity $v_j \in \mathcal{V}$ given the head entity $v_i \in \mathcal{V}$ and the relation $r \in \mathcal{R}$, i.e., $(v_i, r, ?)$; and (3) predict the missing head entity $v_i \in \mathcal{V}$ given the relation $r \in \mathcal{R}$ and the tail entity $v_j \in \mathcal{V}$, i.e., $(?, r, v_j)$. Typically in the literature (Galkin et al., 2021; Chepurova et al., 2023) the problem of predicting the head node is converted to a tail node prediction problem by using the inverse relationship, i.e., (v_j, r^{-1}, v_i). In our experiments, we focus on node prediction rather than relation prediction. We consider the graphs where the relations in the test graph dataset is a subset of the relations in the trainning graph dataset.

Transductive vs inductive link prediction KGC tasks can vary based on the type of knowledge graph: inductive or transductive. In a transductive setting, the task is formulated as follows: given a training knowledge graph $\mathcal{G}_{\text{train}} = (\mathcal{V}, \mathcal{R}, \mathcal{T}_{\text{train}})$, train the model to predict on the inference triplets $(v_i, r, v_j) \in \mathcal{T}_{\text{inference}}$, where $v_i, v_j \in \mathcal{V}$, i.e., nodes in inference triplets are present in the training graph. In inductive KGC, the model is trained on $\mathcal{G}_{\text{train}} = (\mathcal{V}_{\text{train}}, \mathcal{R}, \mathcal{T}_{\text{train}})$ and predicts on the inference triplets $(v_i, r, v_j) \in \mathcal{T}_{\text{inference}}$, where $v_i, v_j \notin \mathcal{V}_{\text{train}}$. By nature KGC in the inductive setting is more challenging compared to the transductive setting.

4. Methodology

In this section, we outline our approach to KGC encompassing both transductive and inductive settings.

4.1. Generating ontology

To generate the ontology $\mathcal{O} = (\mathcal{C}, \mathcal{R}, \mathcal{E})$ of a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{T})$, for each relation $r \in \mathcal{R}$, we create two node sets V_i and V_j of length n that contain head and tail nodes $(v_i, v_j) \in \mathcal{V}$ connected by r. We then prmopt GPT-4 model to predict a head category c_i and a tail category c_j for entities in V_i and V_j respectively, given the relation r. Note that this results in $c_i = c_{v_i}, \forall v_i \in \mathcal{V}_i$ and $c_j = c_{v_j}, \forall v_j \in \mathcal{V}_j$. Naively prompting the LLM to create an ontology by providing node pairs corresponds to a given relation to inconsistencies. GPT-4 model can assign synonyms of node categories to nodes of same class connected to different relations. For example GPT-4 can assign node class label *film* to head node of the triplet (*cruel intentions*, *film cast member actor*, *Alaina Reed Hall*) and assign a node class label *movie* to head node of the triplet (*Dear America: Letters from home Vietnam*, *directed by*, *Bill Couturié*). To improve the quality of the created ontology, we adopt an iterative generation approach that incorporates the previously created sub-ontology at each step. This ensures consistency in node class assignments across similar nodes. Furthermore, we add the triplet (c_{v_i}, r, c_{v_j}) to the set $\mathcal{E}$ to associate each relation $r \in \mathcal{R}$ with exactly one pair of node categories (c_{v_i}, c_{v_j}), maintaining a well-structured ontology.

4.2. Link prediction using topology of the ontology

For a triplet $(v_i, r, v_j) \in \mathcal{T}_{\text{test}}$ with a missing tail v_j (or missing head v_i), we use the generated ontology $\mathcal{O}$ to infer the category c_{v_j} of v_j, based on the relation r and the category c_{v_i} of the head. We provide this inference as a hint to the LLM. Consider the example triplet (*Miles Davis*, *died In*, ?). There are multiple choices for the answer: it can be a city, country, hospital, etc. In our method we find the ontology triplet (*musician*, *died In*, *country*) and let the LLM know that the answer is of type *country*. Additionally, we compute ontology paths between c_{v_i} and c_{v_j}, providing these as context to the LLM. For instance, alternative paths between *musician* and *country* could include (*musician*) $\rightarrow$ *part Of Band* $\rightarrow$ (*band*) $\rightarrow$ *conceived In Country* $\rightarrow$ (*country*).

4.3. Link prediction using topology of the graph

Recall that in transdictive setting nodes in the test graph dataset are present in the training graph dataset. For a triplet $(v_i, r, v_j) \in \mathcal{T}_{\text{inference}}$ with a missing tail v_j, we infer its category c_{v_j} using the ontology $\mathcal{O}$. Next, using $\mathcal{V}_{\text{train}}$ and the category c_{v_j}, we create candidate solutions $v_{\text{candidate}_i} \in \mathcal{V}_{\text{train}}$ that are of category c_{v_j} and prompt the LLM to use the list of candidates as an hint for predicting the missing node. However, the number of candidate nodes we extract from $\mathcal{V}_{\text{train}}$ using the information from ontology can be very large and exceed the limit of the context window. In order to address this problem we employ a strategy which maks multiple LLM calls with sublists of the candidate nodes. Each call tasks the model with selecting the answer that most is most likely to be the missing node. The winning candidates from these calls are subsequently aggregated and provided in a final LLM call as hints to predict the ultimate solution.

4.4. Chain-of-thought style reasoning

In our methodology, we employ chain-of-thought (CoT) reasoning to guide the LLM in making informed predictions about missing nodes in the knowledge graph. CoT reasoning involves providing the LLM with a series of intermediate reasoning steps or questions that help break down the problem and lead to the final answer. By incorporating CoT prompts, we encourage the LLM to consider relevant information from the ontology and graph structure, enhancing its ability to make accurate predictions. The CoT prompts are designed to prompt the LLM to reason about the potential missing node based on the available information, such as the ontology paths, graph paths specific to the given triplet, and the ontology hint about the missing node type. This approach helps the LLM make logical connections between the available information and the missing node, ultimately leading to improved performance in knowledge graph completion tasks.

5. Results

In this section, we present our experimental setup, including the datasets used, baselines compared, and evaluation metrics employed. We then discuss the results obtained from our experiments and provide a comprehensive analysis of our findings, highlighting the key insights and implications.

Datasets and ontology We utilized the small and large datasets from the Inductive Link Prediction Challenge (ILPC) 2022 (Galkin et al., 2022a). The ILPC datasets are specifically designed for inductive link prediction, meaning they contain disjoint inference graphs with new, unseen entities. Both the small and large ILPC datasets consist of three subsets: inductive training graph dataset, transductive training graph dataset and inference test set. All the evaluations were done on the inference test set.

To create the ontology, we first combined the inductive and transductive training graphs, resulting in a graph with approximately 900k triplets. Following the methodology described in Section 4.1, for each relation in the resulting graph, we sampled 50 examples that are connected by this relation and prompted OpenAI GPT-4 to create two ontology categories that best describe the head and tail examples. We allowed the LLM the option to select already predicted categories if it found them appropriate. We then classified all triplets connected by the relation using the LLM-predicted head and tail ontology categories. We performed a post verification step which validated that each relation is associated with only one pair of node categories. Details of the datasets and ontology are given in Table 1

Experiment Settings During inference, for each triplet in the test set, we used GPT-4 to predict the head and the tail under the following conditions: (1) no context, (2) hint derived from the topology of the ontology, (3) hints derived from the topology of both ontology and graph.

No context. In no-context setting we provide GPT-4 with the triplet with missing node and promt it to directly predict the missing node. The results for this setting in reported as GPT-4 + vanilla in Table 2.

Ontology hints. In this setting we provide supporting hints constructed from the topology of the ontology. We consider 4 experimental settings under this method. In *GPT-4 + ontology* we provide GPT-4 with the category of the missing node infered from the ontology triplet that corresponds to the given relation relation. Ontology paths are the paths that exists between available node and missing node categories in the ontology, except the given relation. In *GPT-4 + ontology paths* we provide the path details an additional hint and in *GPT-4 + ontology + ontology paths* we provide both category of the missing node and alternate ontology paths as hints.

Ontology and graph hints. In this setting, we leverage the topology of the graph to generate candidate solutions for the missing node and provide them as hints to GPT-4. We consider four experimental settings under this method. In *GPT-4 + candidate solutions*, we infer the category of the missing node using the ontology and provide GPT-4 with a list of candidate nodes from the training graph that belong to the same category. In *GPT-4 + candidate solutions + ontology*, we provide both the candidate solutions and the ontology-inferred category of the missing node as hints. *GPT-4 + candidate solutions + ontology paths* extends the previous setting by including the alternate paths between the available node and the missing node categories in the ontology as additional hints. Finally, in *GPT-4 + candidate solutions + ontology + ontology paths*, we provide GPT-4 with the candidate solutions, the ontology-inferred category of the missing node, and the alternate ontology paths as comprehensive hints to guide the prediction of the missing node.

Candidate node selection. To generate candidate nodes for the missing node in the transductive setting, we leverage the ontology and the graph structure. First, we infer the category of the missing node from the ontology based on the given relation. Then, we identify all the nodes in the graph that belong to the inferred category and are connected to the available node through the given relation. However, due to the limited context window size of GPT-4, processing all candidate nodes simultaneously may not be feasible. To

Table 1. Dataset details for ILPC-small and ILPC-large.

Dataset	ILPC-small			ILPC-large		
	# nodes	# relations	# triplets	# nodes	# relations	# triplets
Inductive training graph	10,230	96	78,616	46,626	130	202,446
Transductive training graph	6,653	96	20,960	29,246	130	77,044
Ontology graph	36	96	96	42	130	130
Inference test graph	6,653	96	2,902	29,246	130	10,184

Table 2. Hits@k results on ILPC-small, and ILPC-large in transductive setting.

Method	ILPC-small			ILPC-large		
	H@1	H@3	H@10	H@1	H@3	H@10
GPT-4	0.132	0.208	0.289	0.146	0.204	0.253
GPT-4 + neighbors (Chepurova et al., 2023)	0.122	0.202	0.288	0.154	0.208	0.273
GPT-4 + candidate solutions	0.172	0.233	0.319	0.177	0.246	0.292
GPT-4 + candidate solutions + ontology	0.173	0.234	0.318	0.176	0.245	0.292
GPT-4 + candidate solutions + ontology paths	**0.174**	**0.237**	**0.322**	**0.178**	**0.251**	**0.300**
GPT-4 + candidate solutions + ontology + ontology paths	**0.174**	0.236	0.317	0.177	0.249	0.297

address this, we employ a batch-wise approach, where we divide the candidate nodes into batches of 2,000. For each batch, we prompt GPT-4 to select the most likely candidate node. After processing all the batches, we compile the selected candidate nodes from each batch and include them as hints for the final prediction of the missing node. By providing GPT-4 with a refined set of candidate nodes, we aim to enhance its ability to accurately predict the missing node in the transductive setting.

Hyperparameters Here we provide the hyper parameters used in our experiments. All the results are provided with GPT-4-32k model with temperature 0.0 and 2000 maximum number of output tokens. For reproducibility of results we have provided all the prompts used in the experiments in the appendix.

Baselines Our first baseline involved implementing a vanilla GPT-4 prediction without any context window, where it was tasked with predicting the correct missing node (head or tail) from each triplet in the test set. Results are reported in Table 2. For our second baseline, we implemented the method proposed by Chepurova et al. (Chepurova et al., 2023). Here, we augmented each triplet in the inference test set with the 1-hop neighbors of the head (or tail) entity obtained from the ILPC transductive train graph. GPT-4 then received this contextual information and was tasked with predicting the missing tail (or head) of the triplet. Results are reported in 2. For the inductive setting where the LLM did not have access to the test entities, we compared our results with two baselines reported by the authors of the ILPC small and large datasets (Galkin et al., 2022a). These baselines evaluate two variants of the NodePiece model (Galkin et al., 2021), a proposed method for inductive graph representation learning. Results are reported in 3

Evaluation Metrics To assess the performance of our knowledge graph completion approach, we employ the widely adopted Hit@k evaluation metric, specifically focusing on Hit@1, Hit@3, and Hit@10. The Hit@k metric measures the accuracy of the model in predicting the correct missing node within the top k ranked candidates. In our evaluation, we consider three different values of k to provide a comprehensive understanding of the model's performance at various levels of precision. Hit@1 represents the strictest evaluation criterion, where the model is considered successful only if the correct missing entity is ranked as the top candidate. This metric assesses the model's ability to accurately identify the most likely answer for a given query. Hit@3 and Hit@10 provide more relaxed evaluation criteria, allowing the correct missing node to be ranked within the top 3 and top 10 candidates, respectively.

5.1. Results and Analysis

In this section, we present a comprehensive analysis of our experimental results and discuss the key insights and implications derived from our findings.

Finding 1: LLMs demonstrate strong performance in knowledge graph completion tasks.

Our experimental results show that GPT-4, even without any additional context or information, performs significantly better than the baselines on the ILPC-small and ILPC-

Table 3. Hits@k results on ILPC-small, and ILPC-large in inductive setting.

Method	ILPC-small			ILPC-large		
	H@1	H@3	H@10	H@1	H@3	H@10
IndNodePiece (Galkin et al., 2022b)	0.007	0.022	0.092	0.037	0.054	0.125
IndNodePieceGNN (Galkin et al., 2022b)	0.076	0.140	0.251	0.032	0.073	0.146
GPT-4 + ontology	0.134	0.207	0.288	0.150	0.202	0.247
GPT-4 + ontology paths	0.135	0.**210**	**0.290**	0.150	**0.208**	**0.256**
GPT-4 + ontology + ontology paths	**0.138**	0.208	0.289	**0.152**	0.205	0.252

Table 4. Hits@k results on ILPC-small dataset with different numbers of candidate nodes in the context. The numbers given in the table are for the case where GPT-4 is prompted to choose one candidate node from each sub candidate node list. Numbers in parentheses represent the case where multiple candidate nodes are selected from each sublist, totaling 100 candidate nodes.

Method	ILPC Small		
	H@1	H@3	H@10
GPT-4 + candidate solution	0.172 (0.196)	0.233 (0.305)	0.319 (0.391)
GPT-4 + candidate solutions + ontology	0.173 (0.187)	0.274 (0.290)	0.318 (0.383)
GPT-4 + candidate solutions + ontology paths	0.174 (0.198)	0.237 (0.295)	0.322 (0.389)
GPT-4 + candidate solutions + ontology + ontology paths	0.174 (0.196)	0.236 (0.294)	0.317 (0.386)

Table 5. Number of accurate head node predictions obtained by direct head node prediction compared to predicting tail node with inverse relation in ILPC-small dataset.

Method	ILPC Small		
	H@1	H@3	H@10
GPT-4	4	9	12
GPT-4 + ontology	4	10	20
GPT-4 + ontology + paths	6	19	20
GPT-4 + ontology + ontology paths	2	7	19

large datasets. As shown in Table 2, GPT-4 obtains Hit@1 scores of 0.132 and 0.146 on the ILPC-small and ILPC-large datasets, respectively. The table 2 further illustrates similar performance with Hit@3 and Hit@10 as well.

Finding 2: Leveraging candidate solutions significantly improves LLM performance.

In the transductive setting, where test nodes are a subset of nodes in training graph, our approach of generating candidate solutions based on the ontology and utilizing LLMs to select the correct answer yields substantial performance gains. As shown in Table 2 GPT-4 + candidate solutions achieves Hit@1 scores of 0.172 and 0.177 on the ILPC-small and ILPC-large datasets, respectively, outperforming the GPT-4 baseline and the GPT-4 + neighbors approach proposed by (Chepurova et al., 2023). Furthermore, incorporating ontology information and ontology paths alongside the candidate solutions leads to even higher performance. However results illustrate that adding both ontology hint and ontology paths does not necessarily offer a performance improvement. The reason for this is that ontology paths already contains the information about the node category of the missing node, which is the piece of information provided in the ontology method.

Finding 3: Incorporating ontology information enhances LLM performance in the inductive setting

By providing GPT-4 with ontology information, such as the category of the answer and paths between the head and tail categories, we observe a consistent improvement in performance across both the ILPC-small and ILPC-large datasets. As shown in Table 3, GPT-4 + ontology achieves Hit@1 scores of 0.134 and 0.150 on the ILPC-small and ILPC-large datasets, respectively, outperforming the baseline GPT-4 model. Similarly, GPT-4 + ontology paths and GPT-4 + ontology + ontology paths demonstrate even higher performance, with Hit@1 scores reaching 0.138 and 0.152 on the ILPC-small and ILPC-large datasets, respectively. These results suggest that incorporating ontological knowledge and structural information from the graph can guide LLMs towards more accurate predictions in the inductive setting.

Finding 4: Our approach outperforms state-of-the-art baselines in both inductive and transductive settings

Comparing our results to the state-of-the-art baselines, IndNodePiece and IndNodePieceGNN (Galkin et al., 2021). We observe that our approach significantly outperforms these methods in both the inductive and transductive settings. As shown in Table 2, our best-performing model, GPT-4 + ontology + ontology paths, achieves Hit@1 scores of 0.138 and 0.152 on the ILPC-small and ILPC-large datasets, respectively, in the inductive setting, surpassing the baselines by a wide margin. Similarly, in the transductive setting, our approach demonstrates superior performance, with GPT-4 + candidate solutions + ontology paths achieving Hit@1 scores of 0.174 and 0.178 on the ILPC-small and ILPC-large datasets, respectively. These results highlight the effectiveness of our approach in leveraging LLMs and incorporating topological information for knowledge graph completion.

Finding 5: Increasing the number of candidate nodes given in the context improves the performance.

Table 4 presents the performance of our proposed methods on the ILPC-small dataset when using different numbers of candidate nodes are given in the context window. The numbers outside the parentheses represent the case where we choose a single candidate node from each sublist of the original candidate nodes. In contrast, the numbers inside the parentheses correspond to the case where we select multiple candidate nodes from each sublist, such that the total number of chosen candidate nodes is 100. We observe a consistent improvement in performance when increasing the number of candidate nodes provided in the context. For instance, GPT-4 + candidate solutions achieves Hit@1, Hit@3, and Hit@10 scores of 0.172, 0.233, and 0.319, respectively, when using a single candidate node from each sublist. However, when using multiple candidate nodes totaling 100, the scores increase to 0.196, 0.305, and 0.391, respectively. This trend holds for all the other methods as well, demonstrating that providing a larger number of relevant candidate nodes in the context enhances the LLM's ability to accurately predict the missing node.

Finding 6: Predicting head nodes directly yields similar performance to predicting tail nodes with inverse relations.

In the literature, the problem of predicting head nodes is typically converted to a tail node prediction problem by considering the inverse of the relation, denoted as r^{-1} which constructed by adding the word *inverse* as a prefix to the original relation. For a triplet (v_i, r, v_j), predicting the head node v_i is transformed into predicting the tail node in (v_j, r^{-1}, ?). However, in our experiments, we compared the results of directly predicting the head node (?, r, v_j) with the inverse relation approach (v_j, r^{-1}, ?) using GPT-4 with the proposed methods. Interestingly, we found no significant difference in performance between these two approaches when using GPT-4 with our proposed methods. To validate this finding, we conducted experiments on the ILPC-small dataset using both the direct head node prediction and the inverse relation approach with vanilla GPT-4 prompting and ontology based prompting. The results consistently showed that the performance of direct head node prediction was comparable to that of tail node prediction with inverse relations across all methods. Table 5 presents the number of correct head node predictions made by direct head node prediction method compared to using inverse relation head node prediction method out of the total 2902 head node predictions.

6. Discussion and conclusions

In this paper, we have presented a novel approach to KGC leveraging LLMs in both inductive and transductive settings. Our method introduces a generative ontology creation process using LLMs, which extracts structured knowledge directly from raw knowledge graph data. This ontology serves as a foundation, providing crucial cues for inferring missing node categories and pathways between ontology entities.

In the inductive setting, our approach utilizes the ontology and category inference to enhance missing node prediction, demonstrating significant improvements in predictive accuracy without requiring additional training. Additionally, in the transductive setting,our method effectively identifies candidate solutions for triplets using the ontology and selects the correct solution through LLM inference.

These preliminary results are promising and indicate the potential of our approach. Moving forward, our research directions include incorporating important pathways between nodes, exploring online learning techniques to adapt to evolving knowledge graphs with varying triplet probabilities, integrating additional information from diverse external sources into the graph, and expanding our experimental validation to include more datasets.

7. Limitations

The ontology constructed in our method operates under a closed world assumption, where no new entities are added after the initial ontology creation phase. This assumption may limit the adaptability of our approach in dynamic knowledge graph environments where new entities frequently emerge. Looking ahead, addressing this limitation will guide us in enhancing the robustness and applicability of our framework. Another limitation of our approach is that its performance may be impacted by the density of the graph dataset. The effectiveness of leveraging ontology paths relies on the presence of a rich set of relationships and connections within the graph. In cases where the knowledge graph is sparse and lacks a sufficient number of connections between nodes, the ontology paths may not provide significant support for link prediction.

References

Geonames. https://www.geonames.org/. Accessed: 2024-05-23.

Medicomp systems medcin. https://medicomp.com/medcin, 2024. Accessed: 2024-05-23.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., and Duan, Z. Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456, 2020.

Chepurova, A., Bulatov, A., Kuratov, Y., and Burtsev, M. Better together: Enhancing generative knowledge graph completion with language models and neighborhood information. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Chicaiza, J. and Valdiviezo-Diaz, P. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information*, 12(6):232, 2021.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601–610, 2014.

Galkin, M., Denis, E., Wu, J., and Hamilton, W. L. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. *arXiv preprint arXiv:2106.12144*, 2021.

Galkin, M., Berrendorf, M., and Hoyt, C. T. An Open Challenge for Inductive Link Prediction on Knowledge Graphs. mar 2022a. URL http://arxiv.org/abs/2203.01520.

Galkin, M., Berrendorf, M., and Hoyt, C. T. An open challenge for inductive link prediction on knowledge graphs. *arXiv preprint arXiv:2203.01520*, 2022b.

Gemini Team, e. a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

Giglou, H. B., D'Souza, J., and Auer, S. Llms4ol: Large language models for ontology learning, 2023.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., and Zimmermann, A. Knowledge graphs. *ACM Comput. Surv.*, 54(4), jul 2021. ISSN 0360-0300. doi: 10.1145/3447772. URL https://doi.org/10.1145/3447772.

Kazemi, S. M. and Poole, D. Simple embedding for link prediction in knowledge graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 4289–4300, Red Hook, NY, USA, 2018. Curran Associates Inc.

Kochsiek, A., Saxena, A., Nair, I., and Gemulla, R. Friendly neighbors: Contextualized sequence-to-sequence link prediction. *arXiv preprint arXiv:2305.13059*, 2023.

Kommineni, V. K., König-Ries, B., and Samuel, S. From human experts to machines: An llm supported approach to ontology and knowledge graph construction, 2024.

Li, J., Sun, A., Han, J., and Li, C. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2022. doi: 10.1109/TKDE.2020.2981314.

Mazumder, S., Liu, B., Wang, S., and Ma, N. Lifelong and interactive learning of factual knowledge in dialogues. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5903. URL http://dx.doi.org/10.18653/v1/W19-5903.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

Mohamed, H. A., Pilutti, D., James, S., Del Bue, A., Pelillo, M., and Vascon, S. Locality-aware subgraphs for inductive link prediction in knowledge graphs. *Pattern Recognition Letters*, 167:90–97, 2023. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2023.02.004. URL https://www.sciencedirect.com/science/article/pii/S0167865523000314.

Neelakantan, A., Roth, B., and McCallum, A. Compositional vector space models for knowledge base completion, 2015.

Nguyen, D. Q., Tong, V., Phung, D., and Nguyen, D. Q. Node co-occurrence based graph neural networks for knowledge graph link prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, pp. 1589–1592, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3502183. URL https://doi.org/10.1145/3488560.3502183.

OpenAI, e. a. Gpt-4 technical report, 2024.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–20, 2024. ISSN 2326-3865. doi: 10.1109/tkde.2024.3352100. URL http://dx.doi.org/10.1109/TKDE.2024.3352100.

Pawar, S., Palshikar, G. K., and Bhattacharyya, P. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*, 2017.

Saxena, A., Kochsiek, A., and Gemulla, R. Sequence-to-sequence knowledge graph completion and question answering. *arXiv preprint arXiv:2203.10321*, 2022.

Shen, T., Zhang, F., and Cheng, J. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, 255:109597, 2022.

Shi, B. and Weninger, T. Open-world knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Song, Y., Li, W., Dai, G., and Shang, X. Advancements in complex knowledge graph question answering: a survey. *Electronics*, 12(21):4395, 2023.

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. Representing text for joint embedding of text and knowledge bases. In Màrquez, L., Callison-Burch, C., and Su, J. (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1174. URL https://aclanthology.org/D15-1174.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., and Tang, J. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

Yan, J., Wang, C., Cheng, W., Gao, M., and Zhou, A. A retrospective of knowledge graphs. *Frontiers of Computer Science*, sep 2016. doi: 10.1007/s11704-016-5228-9. URL https://doi.org/10.1007/s11704-016-5228-9.

Yani, M. and Krisnadhi, A. A. Challenges, techniques, and trends of simple knowledge graph question answering: a survey. *Information*, 12(7):271, 2021.

Zha, H., Chen, Z., and Yan, X. Inductive relation prediction by bert. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 5923–5931, 2022.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/53f0d7c537d99b3824f0f99d62ea2428-Paper.pdf.

Zhang, Q., Wang, R., Yang, J., and Xue, L. Knowledge graph embedding by translating in time domain space for link prediction. *Knowledge-Based Systems*, 212:106564, 2021. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2020.106564. URL https://www.sciencedirect.com/science/article/pii/S0950705120306936.

Zhang, Z., Cai, J., Zhang, Y., and Wang, J. Learning hierarchy-aware knowledge graph embeddings for link prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):3065–3072, Apr. 2020. doi: 10.1609/aaai.v34i03.5701. URL https://ojs.aaai.org/index.php/AAAI/article/view/5701.

Zou, L., Özsu, M. T., Chen, L., Shen, X., Huang, R., and Zhao, D. gstore: a graph-based sparql query engine. *The VLDB journal*, 23:565–590, 2014.

A. Prompts used for results generation

Vannila Prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Your task:

Triplet with missing node: {triplet}

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Ontology Prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Your task:

Triplet with missing node: {triplet}

Hint about missing node type: The missing node should be of type {type}

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Ontology Paths *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Example:

Triplet with missing node: John Lennon –> born_in –> ?

Available node John Lennon is of node type person.

Graph paths that can be important for filling missing information for a triplet type person –> born_in –> country are [person –> died_in –> country, person –> child_of –> person –> citizen_of –> country].

Chain of thought: This can be a list of questions that might help predict the missing node: [In which country the person died?, If a person is a child of another person who is a citizen of a certain country, which country is that?].

If John Lennon –> died_in –> United Kingdom, John Lennon –> child_of –> Alfred Lennon –> citizen_of –> United Kingdom then it is likely that John Lennon –> born_in –> United Kingdom

Missing node: United Kingdom

Your task:

Triplet with missing node: {triplet}

Available node {known node} is of node type {type}.

Graph paths that are important for filling missing information for a triplet type are {ontology paths}.

Reason about the missing node using Chain of Thought method.

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Ontology and Ontology paths prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Example:

Triplet with missing node: John Lennon –> born_in –> ?

Available node John Lennon is of node type person.

Graph paths that can be important for filling missing information for a triplet type person –> born_in –> country are [person –> died_in –> country, person –> child_of –> person –> citizen_of –> country].

Chain of thought: This can be a list of questions that might help predict the missing node: [In which country the person died?, If a person is a child of another person who is a citizen of a certain country, which country is that?].

If John Lennon –> died_in –> United Kingdom, John Lennon –> child_of –> Alfred Lennon –> citizen_of –> United Kingdom then it is likely that John Lennon –> born_in –> United Kingdom

Hint about missing node type: The missing node should be of type country

Missing node: United Kingdom

Your task:

Triplet with missing node: {triplet}

Available node {known node} is of node type {type}.

Graph paths that are important for filling missing information for a triplet type are {ontology paths}.

Reason about the missing node using Chain of Thought method.

Hint about the missing node type: The missing node should be of type {type}

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Neighbor Prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Your task:

Triplet with missing node: {triplet}

1-hop neighbours of the available node {known node} are given along with their relations as a list: {neighbours}. This does not mean that the missing node is in this list. It is just a hint to help you predict the missing node.

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Candidate nodes prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Your task:

Triplet with missing node: {triplet}

Hint about missing node: The missing node should be of type {type}. Potential candidate nodes for the missing node are {data}. This does not mean that missing node is always in the provided list. It is a hint to help you predict the missing node.

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Candidate nodes with Ontology hint prompt Same as Candidate nodes prompt with
Hint about the missing node type: The missing node should be of type {type}

Candidate nodes with Ontology paths prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Example:

Triplet with missing node: John Lennon –> born_in –> ?

Available node John Lennon is of node type person.

Graph paths that can be important for filling missing information for a triplet type person –> born_in –> country are [person –> died_in –> country, person –> child_of –> person –> citizen_of –> country].

Chain of thought: This can be a list of questions that might help predict the missing node: [In which country the person died?, If a person is a child of another person who is a citizen of a certain country, which country is that?].

If John Lennon –> died_in –> United Kingdom, John Lennon –> child_of –> Alfred Lennon –> citizen_of –> United Kingdom then it is likely that John Lennon –> born_in –> United Kingdom

Hint about missing node type: The missing node should be of type country

Missing node: United Kingdom

Your task:

Triplet with missing node: {triplet}

Available node {known node} is of node type {type}.

Hint about missing node: The missing node should be of type {type}. Potential candidate nodes for the missing node are {data}. This does not mean that missing node is always in the provided list. It is a hint to help you predict the missing node.

Graph paths that are important for filling missing information for a triplet type are {ontology paths}.

Reason about the missing node using Chain of Thought method.

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Candidate nodes with Ontology paths and hit prompt Same as Candidate nodes with Ontology paths prompt with *Hint about the missing node type: The missing node should be of type {type}*

Candidate nodes with ontology hint and graph paths prompt *You will receive a triplet with a missing node. Given triplet can be of the form available node –> relation –> ? or ? –> relation –> available node. Your task is predicting the missing node. Alongside, you will get a hint about the type of answer that is correct, and you might receive additional relevant information to aid your prediction. Please respond only with the missing node, without including the head node or the relation.*

Example:

Triplet with missing node: John Lennon –> born_in –> ?

Available node John Lennon is of node type person.

Graph paths that can be important for filling missing information for triplet John Lennon –> born_in –> ? are [John Lennon –> died_in –> United Kingdom, John Lennon –> child_of –> Alfred Lennon –> citizen_of –> United Kingdom].

Chain of thought:

If John Lennon died_in United Kingdom, and John Lennon is a child_of Alfred Lennon who is a citizen_of United Kingdom then it is likely that John Lennon –> born_in –> United Kingdom

Hint about missing node type: The missing node should be of type country

Missing node: United Kingdom

Your task:

Triplet with missing node: {triplet}

Hint about missing node: The missing node should be of type {type}. Potential candidate nodes for the missing node are {data}. This does not mean that missing node is always in the provided list. It is a hint to help you predict the missing node.

Graph paths that are important for filling missing information for triplet {triplet} are {graph paths}.

Reason about the missing node using Chain of Thought method.

Hint about the missing node type: The missing node should be of type {type}

Please provide a list of 10 candidate nodes for the missing node. Answer should be of the format ['candidate_node_1', 'candidate_node_2', ..., 'candidate_node_10']. The list of candidate nodes should be in order of most probably candidate node to least probable candidate node.

Missing candidate nodes:

Ontology Prompt *I will provide you with a relation and two data pairs. Your task is to determine the specific ontology node classes for the entities in these data pairs that are connected by the specified relation.*

Strict Requirements: 1) All node classes must be written in lowercase. 2) Connect words within node classes using underscores. 3) The relation provided must not be altered in your response. 4) Provide a single response for all data pairs, formatted as follows: ['head node class', 'tail node class', 'relation']. 5) Ensure your answer strictly adheres to this format: ['head node class', 'tail node class', 'relation']. Do not include any additional text or explanation.

Soft Requirements: 1) Refer to the existing node classes: {ontology_categories}. You may reuse these if they accurately describe the data pairs. If not, provide a more suitable classification. 2) Avoid using generic terms like 'person'. Instead, use more specific classifications such as 'film_producer' or 'play_writer' where applicable.

Example 1: Relation: 'was born in' Data Pairs: ['(John Lennon, United Kingdom)', '(Miles Davis, United States)'] Answer: ['musician', 'country', 'was born in']

Example 2: Relation: 'directed by' Data Pairs: ['(Inception, Christopher Nolan)', '(Titanic, James Cameron)'] Answer: ['film', 'film_director', 'directed by']

Your turn: Relation: '{relation}' Data Pairs: '{data_pairs}' Answer:

B. Paths identified in the ontology

B.1. Alternate ontology paths for relations in ILPC-small dataset

In this section we provide the alternate ontology paths for a given ontology triplet

Individual → medical condition → medical condition

- Individual → cause of death → medical condition

Individual → place of birth → city

- Individual → employer → university or organization → headquarters location → city
- Individual → place of death → city
- Individual → residence → city

Individual → part of → organization

- Individual → member of → organization

Individual → residence → city

- Individual → employer → university or organization → headquarters location → city
- Individual → place of death → city
- Individual → place of birth → city

Individual → languages spoken, written or signed → language

- Individual → employer → university or organization → located in the administrative territorial entity → city or country → official language → language

Individual → member of → organization

- Individual → part of → organization

Individual → place of death → city

- Individual → employer → university or organization → headquarters location → city
- Individual → residence → city
- Individual → place of birth → city

University or organization → headquarters location → city

- University or organization → located in the administrative territorial entity → city or country → named after → individual → place of death → city
- University or organization → located in the administrative territorial entity → city or country → named after → individual → residence → city
- University or organization → located in the administrative territorial entity → city or country → named after → individual → place of birth → city

Individual → cause of death → medical condition

- Individual → medical condition → medical condition

City or country → official language → language

- City or country → named after → individual → languages spoken written or signed → language

B.2. Alternate ontology paths for relations in ILPC-large dataset

Individual → field of work → field of work

- Individual → is the study of → field of work

Location → country → country

- Location → named after → individual → country of citizenship → country
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → founded by → individual → country of citizenship → country
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chief executive officer → individual → country of citizenship → country
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chairperson → individual → country of citizenship → country

Location → has use → sport

- Location → named after → individual → sport → sport
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → founded by → individual → sport → sport
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chief executive officer → individual → sport → sport
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chairperson → individual → sport → sport

Individual → languages spoken, written or signed → language

- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Individual → part of → organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → official language → language

Individual → movement → genre

- Individual → genre → genre

Individual → employer → educational institution

- Individual → educated at → educational institution

Organization → chairperson → individual

- Organization → founded by → individual
- Organization → chief executive officer → individual
- Organization → airline hub → location → named after → individual

Organization → chief executive officer → individual

- Organization → founded by → individual
- Organization → chairperson → individual
- Organization → airline hub → location → named after → individual

Individual → educated at → educational institution

- Individual → employer → educational institution

Creative work → country of origin → country

- Creative work → creator → individual → place of burial → location → country → country
- Creative work → creator → individual → country of citizenship → country
- Creative work → creator → individual → part of → organization → airline hub → location → country → country
- Creative work → publisher → organization → founded by → individual → place of burial → location → country → country
- Creative work → publisher → organization → founded by → individual → country of citizenship → country
- Creative work → publisher → organization → chief executive officer → individual → place of burial → location → country → country
- Creative work → publisher → organization → chief executive officer → individual → country of citizenship → country

- Creative work → publisher → organization → chairperson → individual → place of burial → location → country → country
- Creative work → publisher → organization → chairperson → individual → country of citizenship → country
- Creative work → publisher → organization → airline hub → location → country → country
- Creative work → publisher → organization → airline hub → location → named after → individual → country of citizenship → country
- Creative work → narrative location → location → country → country
- Creative work → narrative location → location → named after → individual → country of citizenship → country
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → founded by → individual → country of citizenship → country
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chief executive officer → individual → country of citizenship → country
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chairperson → individual → country of citizenship → country

Organization → headquarters location → city

- Organization → founded by → individual → place of death → city
- Organization → founded by → individual → place of birth → city
- Organization → founded by → individual → residence → city
- Organization → founded by → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → capital → city
- Organization → chief executive officer → individual → place of death → city
- Organization → chief executive officer → individual → place of birth → city
- Organization → chief executive officer → individual → residence → city
- Organization → chief executive officer → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → capital → city
- Organization → chairperson → individual → place of death → city
- Organization → chairperson → individual → place of birth → city
- Organization → chairperson → individual → residence → city
- Organization → chairperson → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → capital → city
- Organization → airline hub → location → named after → individual → place of death → city
- Organization → airline hub → location → named after → individual → place of birth → city
- Organization → airline hub → location → named after → individual → residence → city
- Organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → capital → city

Individual → residence → city

- Individual → place of death → city
- Individual → place of birth → city
- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → capital → city
- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → headquarters location → city
- Individual → part of → organization → headquarters location → city
- Individual → part of → organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → capital → city

Individual → place of birth → city

- Individual → place of death → city
- Individual → residence → city
- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → capital → city

- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → headquarters location → city
- Individual → part of → organization → headquarters location → city
- Individual → part of → organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → capital → city

Creative work → narrative location → location

- Creative work → creator → individual → place of burial → location
- Creative work → creator → individual → part of → organization → airline hub → location
- Creative work → publisher → organization → founded by → individual → place of burial → location
- Creative work → publisher → organization → chief executive officer → individual → place of burial → location
- Creative work → publisher → organization → chairperson → individual → place of burial → location
- Creative work → publisher → organization → airline hub → location

Individual → place of burial → location

- Individual → part of → organization → airline hub → location

Individual → country of citizenship → country

- Individual → place of burial → location → country → country
- Individual → part of → organization → airline hub → location → country → country

Creative work → publisher → organization

- Creative work → creator → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization
- Creative work → creator → individual → part of → organization
- Creative work → narrative location → location → named after → individual → part of → organization
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization

Administrative territorial entity → capital → city

- Administrative territorial entity → legislative body → organization → headquarters location → city
- Administrative territorial entity → legislative body → organization → founded by → individual → place of death → city
- Administrative territorial entity → legislative body → organization → founded by → individual → place of birth → city
- Administrative territorial entity → legislative body → organization → founded by → individual → residence → city
- Administrative territorial entity → legislative body → organization → chief executive officer → individual → place of death → city
- Administrative territorial entity → legislative body → organization → chief executive officer → individual → place of birth → city
- Administrative territorial entity → legislative body → organization → chief executive officer → individual → residence → city
- Administrative territorial entity → legislative body → organization → chairperson → individual → place of death → city
- Administrative territorial entity → legislative body → organization → chairperson → individual → place of birth → city
- Administrative territorial entity → legislative body → organization → chairperson → individual → residence → city
- Administrative territorial entity → legislative body → organization → airline hub → location → named after → individual → place of death → city
- Administrative territorial entity → legislative body → organization → airline hub → location → named after → individual → place of birth → city
- Administrative territorial entity → legislative body → organization → airline hub → location → named after → individual → residence → city

Location → named after → individual

- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → founded by → individual
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chief executive officer → individual
- Location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chairperson → individual

Organization → founded by → individual

- Organization → chief executive officer → individual
- Organization → chairperson → individual
- Organization → airline hub → location → named after → individual

Organization → airline hub → location

- Organization → founded by → individual → place of burial → location
- Organization → chief executive officer → individual → place of burial → location
- Organization → chairperson → individual → place of burial → location

Individual → genre → genre

- Individual → movement → genre

Administrative territorial entity → official language → language

- Administrative territorial entity → legislative body → organization → founded by → individual → languages spoken, written or signed → language
- Administrative territorial entity → legislative body → organization → chief executive officer → individual → languages spoken, written or signed → language
- Administrative territorial entity → legislative body → organization → chairperson → individual → languages spoken, written or signed → language
- Administrative territorial entity → legislative body → organization → airline hub → location → named after → individual → languages spoken, written or signed → language

Individual → sport → sport

- Individual → place of burial → location → has use → sport
- Individual → part of → organization → airline hub → location → has use → sport

Individual → unmarried partner → individual

Creative work → creator → individual

- Creative work → publisher → organization → founded by → individual
- Creative work → publisher → organization → chief executive officer → individual
- Creative work → publisher → organization → chairperson → individual
- Creative work → publisher → organization → airline hub → location → named after → individual
- Creative work → narrative location → location → named after → individual
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → founded by → individual
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chief executive officer → individual
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chairperson → individual

Individual → occupation → occupation

- Individual → n/a → occupation

Individual → is the study of → field of work

- Individual → field of work → field of work

Individual → place of death → city

- Individual → place of birth → city
- Individual → residence → city
- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → capital → city
- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → headquarters location → city
- Individual → part of → organization → headquarters location → city
- Individual → part of → organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → capital → city

Creative work → language of work or name → language

- Creative work → creator → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → creator → individual → languages spoken, written or signed → language
- Creative work → creator → individual → part of → organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → publisher → organization → founded by → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → publisher → organization → founded by → individual → languages spoken, written or signed → language
- Creative work → publisher → organization → chief executive officer → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → publisher → organization → chief executive officer → individual → languages spoken, written or signed → language
- Creative work → publisher → organization → chairperson → individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → publisher → organization → chairperson → individual → languages spoken, written or signed → language
- Creative work → publisher → organization → airline hub → location → named after → individual → languages spoken, written or signed → language
- Creative work → publisher → organization → airline hub → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → narrative location → location → named after → individual → languages spoken, written or signed → language
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → official language → language
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → founded by → individual → languages spoken, written or signed → language
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chief executive officer → individual → languages spoken, written or signed → language
- Creative work → narrative location → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization → chairperson → individual → languages spoken, written or signed → language

Individual → part of → organization

- Individual → place of burial → location → located in the administrative territorial entity → administrative territorial entity → legislative body → organization

Individual → n/a → occupation

- Individual → occupation → occupation