A Unified Taxonomy-Guided Instruction Tuning Framework for Entity Set Expansion and Taxonomy Expansion

Yanzhen Shen, Yu Zhang, Yunyi Zhang, Jiawei Han

Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA
{yanzhen4, yuz9, yzhan238, hanj}@illinois.edu

Abstract

Scientific taxonomy plays a crucial role in organizing and structuring scientific knowledge across various fields like Medical Science and Computer Science. With the rapid advancement of scientific research and the emergence of new scientific concepts, people have also sought to automatically populate an existing taxonomy. Entity set expansion, taxonomy expansion, and seed-guided taxonomy construction are three representative tasks that can be applied to automatic taxonomy construction. Previous studies view them as three separate tasks. Therefore, their proposed techniques usually work for one specific task only, lacking generalizability and a holistic perspective. In this paper, we aim at a unified solution to the three tasks. To be specific, we identify two common skills needed for entity set expansion, taxonomy expansion, and seed-guided taxonomy construction: finding "siblings" and finding "parents". We propose a taxonomy-guided instruction tuning framework to teach a large language model to generate siblings and parents for query entities, where the joint pre-training process facilitates the mutual enhancement of the two skills. Extensive experiments on multiple benchmark datasets demonstrate the efficacy of our proposed TAXOINSTRUCT framework, which outperforms task-specific baselines across all three tasks.

Introduction

Entities play a fundamental role in text mining and natural language processing, benefiting a wide spectrum of tasks such as semantic search (Shen et al. 2018b), question answering (Christmann, Saha Roy, and Weikum 2022), and text generation (Li et al. 2021). In scientific research, to better describe the semantics of scientific entities, taxonomies are constructed in various fields, including Medical Science (Coletti and Bleich 2001), Environment (Bordea, Lefever, and Buitelaar 2016), and Computer Science (Shen et al. 2018a), to characterize the parent-child relationship between entities. In many cases, taxonomies are initially curated by domain experts. However, because of the constant and rapid emergence of novel concepts, automatically enriching a taxonomy with new entities becomes necessary to ensure its freshness and completeness. To this end, previous studies have considered three representative tasks for incorporating new entities into existing knowledge.

(1) Entity Set Expansion (Wang and Cohen 2007; Rong et al. 2016; Shen et al. 2017): Given a set of entities belonging to a certain semantic class, the task is to find more entities also in that class. For example, if there are seed entities {*Database*, *Information Retrieval*, *Operating System*}, an entity set expansion algorithm should return other computer science subfields such as *Data Mining* and *Human-Computer Interaction*. From the taxonomy perspective, this task can be viewed as finding "**siblings**" of existing entities.

(2) Taxonomy Expansion (Shen et al. 2020b; Yu et al. 2020; Zeng et al. 2021): The task aims to insert a provided new entity into an existing taxonomy by finding its most suitable "**parents**". For instance, suppose the existing taxonomy has the root node *Scientific Fields* and its children *Computer Science*, *Mathematics*, *Physics*, and *Chemistry*. Given a new concept *Data Mining*, a taxonomy expansion model should put it as a child of *Computer Science*.

(3) Seed-Guided Taxonomy Construction (Shen et al. 2018a): Given a seed taxonomy with a small number of entities, the task is to construct a more comprehensive taxonomy containing the seed taxonomy. For example, if the input includes *Computer Science*, *Chemistry*, and several of their subfields (e.g., *Data Mining* and *Organic Chemistry*), the expected output should be a taxonomy containing more scientific fields (e.g., *Mathematics* and *Physics*) and subfields (e.g., *Database*, *Algebra*, and *Astrophysics*), with their parent-child edges specified. To approach this problem, we can first discover new entities at each layer and then figure out the parent-child edges between entities from adjacent layers. Essentially, this can be viewed as pipelining the steps of finding "**siblings**" and finding "**parents**".

As we can clearly see, all the three aforementioned tasks can be cast as finding entities that have a specific type of relationship with the given entities: entity set expansion can be viewed as finding "siblings"; taxonomy expansion relies on finding "parents"; seed-guided taxonomy construction is a combination of both. However, existing studies always focus on one of the three tasks and propose task-specific techniques, with less concern for their commonalities. Intuitively, finding "siblings" and finding "parents" can mutually benefit each other. For instance, knowing that *Data Mining* has siblings *Database* and *Information Retrieval* helps us predict the parent of *Data Mining* to be *Computer Sci-*

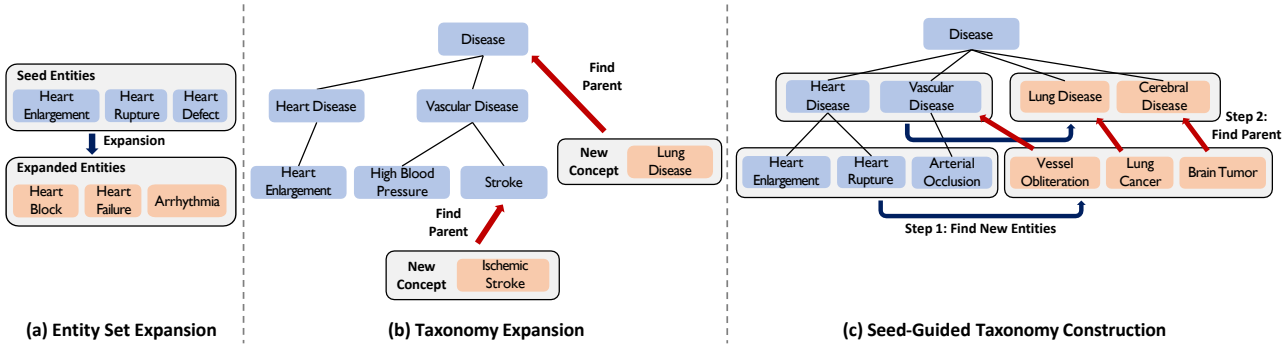

Figure 1: Illustrations of the three tasks.

ence, and vice versa. Once the accuracies of sibling prediction and parent prediction are both improved, by taking them as building blocks, the three tasks can be better solved in a holistic and unified way.

Contributions. Inspired by the idea above, in this paper, we aim at a unified framework to tackle entity set expansion, taxonomy expansion, and seed-guided taxonomy construction simultaneously. To be specific, we leverage existing taxonomies (e.g., the Comparative Toxicogenomics Database (Davis et al. 2022)) as rich sources of sibling-sibling and parent-child relationships to pre-train a model for sibling finding and parent finding. The pre-trained model can be further fine-tuned on domain-specific data (e.g., parent-child pairs in the input taxonomy for the taxonomy expansion task) to perform downstream tasks. To implement this framework, we exploit the ability of large language models (LLMs) (Achiam et al. 2023; Touvron et al. 2023) to follow human instructions (Wei et al. 2022; Ouyang et al. 2022). Our proposed TAXOINSTRUCT framework utilizes task-specific instructions to teach an LLM the skills of generating sibling entities and the parent entity for one or more query entities. The joint pre-training process facilitates the mutual enhancement of the two skills.

To examine the efficacy of TAXOINSTRUCT, we conduct comprehensive experiments on 4 scientific domain and 2 general domain benchmark datasets of entity set expansion, taxonomy expansion, and seed-guided taxonomy construction. Experimental results demonstrate that TAXOINSTRUCT, as a unified framework, significantly outperforms competitive task-specific baselines in all three tasks. We also analyze the performance of TAXOINSTRUCT when different LLM backbones are plugged in, demonstrating that the effectiveness of TAXOINSTRUCT is generic and does not reply on a specific choice of the LLM.

Problem Definition

In this section, we formally introduce the three representative tasks for populating a taxonomy with new entities.

Entity Set Expansion. As shown in Figure 1(a), given a few example entities (also known as "seeds"), the entity set expansion task (Wang and Cohen 2007; Rong et al. 2016; Shen et al. 2017) aims to find a set of "sibling" entities that belong to the same semantic class as the seeds. Formally, we have the following task definition.

Definition 1 (Entity Set Expansion) *Given a small set of seed entities $\mathcal{S} = \{s_1, s_2, ..., s_M\}$, the task is to discover more entities $\mathcal{S}^+ = \{s_{M+1}, s_{M+2}, ..., s_{M+N}\}$, where $s_1, s_2, ..., s_{M+N}$ fall into the same semantic category.*

Taxonomy Expansion. As shown in Figure 1(b), given an existing taxonomy and a set of new entities, taxonomy expansion (Shen et al. 2020b; Yu et al. 2020; Zeng et al. 2021) aims at inserting the new entities into the taxonomy. This process is facilitated by finding a proper "parent" node in the existing taxonomy for each new entity. Formally,

Definition 2 *(Taxonomy Expansion) Given an existing taxonomy $\mathcal{T}$ (which contains a set of entities $\mathcal{S}$ and the parent-child relationship between the entities $\textsf{Parent}(\cdot) : \mathcal{S} \rightarrow \mathcal{S} \cup \{ROOT\}$) and a set of new entities $\mathcal{S}^+$, the task is to expand the taxonomy to a more complete one $\mathcal{T}^+$ with entities $\mathcal{S} \cup \mathcal{S}^+$ and the parent-child relationship $\textsf{Parent}^+(\cdot) : \mathcal{S} \cup \mathcal{S}^+ \rightarrow \mathcal{S} \cup \{ROOT\}$.*

Seed-Guided Taxonomy Construction. As shown in Figure 1(c), seed-guided taxonomy construction (Shen et al. 2018a) deals with the case where we need to first find a set of new entities to be inserted to the taxonomy and then find the proper parent node for each new entity.

Definition 3 *(Seed-Guided Taxonomy Construction) Given a small set of seeds that form a tree structure $\mathcal{T} = (\mathcal{S}_0, \mathcal{S}_1, ..., \mathcal{S}_L)$, where $\mathcal{S}_0 = \{s_{\text{ROOT}}\}$ contains the root node, $\mathcal{S}_l$ $(1 \leq l \leq L)$ denotes the set of seeds at layer l, and the parent-child relationship is characterized by a mapping function $\textsf{Parent}(\cdot) : \mathcal{S}_l \rightarrow \mathcal{S}_{l-1}$, the task aims to discover more entities at each level (denoted by the sets $\mathcal{S}_1^+, ..., \mathcal{S}_L^+$, where entities in $\mathcal{S}_l$ and $\mathcal{S}_l^+$ belong to the same semantic class) and predict their parent-child relationship (characterized by $\textsf{Parent}^+(\cdot) : \mathcal{S}_l \cup \mathcal{S}_l^+ \rightarrow \mathcal{S}_{l-1} \cup \mathcal{S}_{l-1}^+$).*

We need to clarify two key differences between taxonomy expansion (Definition 2) and seed-guided taxonomy construction (Definition 3): First, taxonomy expansion assumes that the new entities to be inserted into the existing taxonomy are already given, while seed-guided taxonomy construction needs to discover these new entities first. In other words, taxonomy expansion mainly focuses on the task of finding "parents", whereas seed-guided taxonomy construction aims to first predict "siblings" and then predict "parents". Second, because seed-guided taxonomy construction

finds new entities that share the same semantic granularity with the seeds at each layer, it can only make the taxonomy "wider" but not "deeper". By contrast, taxonomy expansion can enrich the taxonomy with entities that are more fine-grained than any input seeds.

The TAXOINSTRUCT Framework

According to the definitions of the three tasks, we notice that they can be reduced to two key challenges – finding "siblings" and finding "parents". Entity set expansion relies on the former; taxonomy expansion relies on the latter; seed-guided taxonomy construction can be viewed as a pipeline of both. Inspired by this, we aim to train a unified model that simultaneously supports these two skills (therefore facilitates all three tasks). To implement this idea, in this section, we propose TAXOINSTRUCT, a unified taxonomy-guided instruction tuning framework.

Instruction Tuning for Entity Set Expansion

Given a set of seeds $\mathcal{S} = \{s_1, s_2, ..., s_M\}$ belonging to the same semantic class, the entity set expansion task enforces two restrictions on the expanded entities $\mathcal{S}^+ = \{s_{M+1}, s_{M+2}, ..., s_{M+N}\}$. First, s_{M+n} $(1 \leq n \leq N)$ can be classified into the same category as $s_1, s_2, ..., s_M$. For example, in Figure 1(a), both *Heart Enlargement* and *Arrhythmia* can be classified into the category *Heart Disease*. Second, s_{M+n} must also share the same granularity with $s_1, s_2, ..., s_M$. For example, although *Congenital Heart Defect* belongs to *Heart Disease* as well, it should not be expanded in Figure 1(a) because it is more fine-grained than the seed *Heart Defect*. These two restrictions are inherently describing the concept of "siblings" in a taxonomy, since "siblings" share the same parent and reside in the same level.

Inspired by this, we formulate the entity set expansion task (from a taxonomy perspective) as finding other siblings of the seed entities. We approach this problem by unleashing LLMs' power of following task-specific instructions (Wei et al. 2022; Ouyang et al. 2022; Wang et al. 2023b). Briefly, given a set of INPUT entities $\mathcal{S} = \{s_1, s_2, ..., s_M\}$ that share the same parent node $\mathsf{Parent}(\mathcal{S})$, we INSTRUCT an LLM (e.g., Llama-3 8B[1]) to generate more children of $\mathsf{Parent}(\mathcal{S})$ in its RESPONSE.

Nevertheless, the parent entity $\mathsf{Parent}(\mathcal{S})$ is not available in the standard entity set expansion task (Rong et al. 2016; Shen et al. 2017; Yu et al. 2019). Thus, we first prompt the LLM to generate the parent entity for the seed set $\mathcal{S}$. Following the (INSTRUCTION, INPUT, RESPONSE) schema of Llama-3, we form the instruction as follows:

> INSTRUCTION: *Given a list of entities, output the most likely parent class for the entity given by user.*
>
> INPUT: *Find the parent class for* $\{s_1, s_2, ..., s_M\}$.
>
> RESPONSE: *The parent class is*

The generated parent entity $\mathsf{Parent}(\mathcal{S})$ is then used to guide the expansion process. Because entity set expansion normally contains a very small number of (e.g., 3) seed entities, it is hard to get sufficient self-supervision from the seeds for further fine-tuning. Therefore, we directly perform inference by leveraging the following instruction:

> INSTRUCTION: *Given a category and an entity set belonging to this category, output other entities belonging to this category and sharing the same granularity as the seeds.*
>
> INPUT: *Find other entities belonging to the category* $\mathsf{Parent}(\mathcal{S})$ *and sharing the same granularity as the seeds* $\{s_1, s_2, ..., s_M\}$.
>
> RESPONSE: *The expanded entities are*

The LLM will generate a set of expanded entities, which we denote as $\mathcal{R} = \{r_1, r_2, ..., r_K\}$. After that, we perform a ranking step to sort these entities. To be specific, we use a moderate-size auxiliary pre-trained language model (e.g., BERT (Devlin et al. 2019)) to compute the similarity score between each generated entity $r \in \mathcal{R}$ and $\mathsf{Parent}(\mathcal{S})$:

$$\mathrm{sim}(r, \mathsf{Parent}(\mathcal{S})) = \cos\Big(\mathrm{PLM}(r), \mathrm{PLM}(\mathsf{Parent}(\mathcal{S}))\Big), \quad (1)$$

where $\mathrm{PLM}(\cdot)$ is the average output token embedding after feeding the entity name into the moderate-size auxiliary pre-trained language model. All entities in $\mathcal{R}$ are then ranked according to $\mathrm{sim}(\cdot, \mathsf{Parent}(\mathcal{S}))$. Afterwards, we add the top-κ entities ($\kappa = 3$ in TAXOINSTRUCT) expanded in the first iteration back to the seed entity list $\mathcal{S}$ and rerun the expansion process with the enriched seed set. This process can be conducted iteratively, following the common practice of previous entity set expansion algorithms (Shen et al. 2017; Zhang et al. 2020). After the final iteration, we rank all seeds and expanded entities (except the original seeds which should not appear in the output) according to $\mathrm{sim}(\cdot, \mathsf{Parent}(\mathcal{S}))$ and obtain a list, $\mathcal{S}^+$, of expanded entities.

Instruction Tuning for Taxonomy Expansion

Taxonomy expansion is a parent-finding task. Given a INPUT entity $s_q \in \mathcal{S}^+$, we INSTRUCT an LLM to identify the correct parent node $\mathsf{Parent}(s_q)$ from a provided list of candidates $\mathcal{S} = \{s_1, s_2, ..., s_M\}$ (i.e., entities in the existing taxonomy). We form the (INSTRUCTION, INPUT, RESPONSE) schema as follows:

> INSTRUCTION: *Given a set of candidate parent classes and an entity, output the most likely parent class for the entity given by user.*
>
> INPUT: *Given candidate parents* $\{s_1, s_2, ..., s_M\}$, *find the parent class for* s_q.
>
> RESPONSE: *The parent class is*

In practice, however, the input taxonomy may contain tens of thousands of entities (Shen et al. 2020b; Zeng et al. 2021). If we include all of them as candidates and put them into the instruction, the LLM may be overwhelmed by the overly large label space and can hardly follow the instruction. To tackle this problem, we utilize a moderate-size auxiliary language model (e.g., BERT (Devlin et al. 2019)) to first retrieve a set of candidates from the taxonomy and thus reduce the label space for the LLM. More specifically, given the query s_q, we select top-U (e.g., $U = 20$) entities $\mathcal{U}_q \subseteq \mathcal{S}$ with the highest similarity to s_q.

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B

$$\mathcal{U}_q = \arg\max_{\mathcal{U}\subseteq\mathcal{S},|\mathcal{U}|=U} \sum_{s\in\mathcal{U}} \cos\Big(\mathrm{PLM}(s_q), \mathrm{PLM}(s)\Big), \quad (2)$$

where $\mathrm{PLM}(\cdot)$ has the same meaning as in Eq. (1). The retrieved subset $\mathcal{U}_q$ will replace the entire candidate list in the INPUT.

Since the input taxonomy contains rich (parent, child) entity pairs, we exploit them to fine-tune the LLM so that it better understands the parent-child relationship and domain knowledge. To be specific, given a node s_i in the input taxonomy and its parent $\mathsf{Parent}(s_i)$, we construct training data in two different ways:

Distinguishing the parent from its siblings: We take the siblings of $\mathsf{Parent}(s_i)$ as candidates and fine-tune the LLM to identify the true parent $\mathsf{Parent}(s_i)$ from $\{\mathsf{Parent}(s_i)\} \cup \mathsf{Sibling}(\mathsf{Parent}(s_i))$.

Distinguishing the parent from semantically similar entities: We use Eq. (2) to find the set of top-U entities $\mathcal{U}_i$ that are closest to s_i. Then, the LLM needs to identify the true parent $\mathsf{Parent}(s_i)$ from the candidates $\{\mathsf{Parent}(s_i)\} \cup \mathcal{U}_i$.

Because the candidates will be sequential in the instruction, to mitigate potential effects of their order, we randomly shuffle the candidate list (i.e., either $\{\mathsf{Parent}(s_i)\} \cup \mathsf{Sibling}(\mathsf{Parent}(s_i))$ or $\{\mathsf{Parent}(s_i)\} \cup \mathcal{U}_i$) V times. Each shuffled list $\mathcal{V}_{i,j}$ will be used to construct an individual tuple of (INSTRUCTION, INPUT, RESPONSE) to fine-tune the LLM. Formally, we have:

> INSTRUCTION: *Given a set of candidate parent classes and an entity, output the most likely parent class for the entity given by user.*
>
> INPUT: *Given candidate parents $\mathcal{V}_{i,j}$, find the parent class for s_i.*
>
> RESPONSE: *The parent class is*

The LLM is fine-tuned via maximizing the following objective function (i.e., the log-likelihood to generate the correct parent entity):

$$\sum_{i=1}^{M}\sum_{j=1}^{V} \log \Pr(\,\mathsf{Parent}(s_i) \mid \text{INSTRUCTION}, \text{INPUT}, \text{RESPONSE}\,). \quad (3)$$

Instruction Tuning for Seed-Guided Taxonomy Construction

According to Definition 3, seed-guided taxonomy construction can be naturally divided into two subtasks: (1) expanding the entity set at each layer to discover new entities (i.e., finding "siblings" and "cousins"[2]) and (2) expanding the taxonomy by finding the proper "parent" for each new entity. These two subtasks bear similarity with entity set expansion and taxonomy expansion, respectively. Therefore, we can adopt similar instructions used in previous sections.

[2]In the first step of seed-guided taxonomy construction, we need to find entities that share the same semantic granularity as the seeds at each layer. They are only required to be the descendants of the root node and may not share the same parent entity with the seeds. Therefore, we aim to discover not only "siblings" but also "cousins" of the seeds here.

Finding "Siblings" and "Cousins". Given the input taxonomy $\mathcal{T} = (\mathcal{S}_0, \mathcal{S}_1, ..., \mathcal{S}_L)$ where $\mathcal{S}_0 = \{s_{\mathrm{ROOT}}\}$ and $\mathcal{S}_l = \{s_{l,1}, s_{l,2}, ..., s_{l,M_l}\}$ $(1 \le l \le L)$, we expand each layer l by using the following (INSTRUCTION, INPUT, RESPONSE) schema:

> INSTRUCTION: *Given a category and an entity set belonging to this category, output other entities belonging to this category and sharing the same granularity as the seeds.*
>
> INPUT: *Find other entities belonging to the category s_{ROOT} and sharing the same granularity as the seeds $\{s_{l,1}, s_{l,2}, ..., s_{l,M_l}\}$.*
>
> RESPONSE: *The expanded entities are*

The major difference between this instruction and that for entity set expansion is that we put s_{ROOT} rather than $\mathsf{Parent}(\mathcal{S}_l)$ into the INPUT to discover not only "siblings" but also "cousins" of $\mathcal{S}_l$. We denote the expanded entities at layer l as $\mathcal{S}_l^+ = \{s_{l,M_l+1}, s_{l,M_l+2}, ..., s_{l,M_l+N_l}\}$ $(1 \le l \le L)$.

Finding "Parents". For each newly discovered entity $s_{l,M_l+n} \in \mathcal{S}_l^+ \backslash \mathcal{S}_l$, we need to insert it into the taxonomy by finding its parent from all entities that are one layer coarser. When $l = 1$, this problem is trivial because the parent is s_{ROOT}. When $l \ge 2$, we consider the following instruction:

> INSTRUCTION: *Given a set of candidate parent classes and an entity, output the most likely parent class for the entity given by user.*
>
> INPUT: *Given candidate parents $\{s_{l-1,1}, s_{l-1,2}, ..., s_{l-1,M_{l-1}+N_{l-1}}\}$, find the parent for s_{l,M_l+n}.*
>
> RESPONSE: *The parent class is*

The major difference between this instruction and that for taxonomy expansion is that the candidate parent list in the INSTRUCTION contains entities at layer $l-1$ only (i.e., $\mathcal{S}_{l-1}^+$) rather than the entire input taxonomy.

In seed-guided taxonomy construction, similar to taxonomy expansion, we are given a taxonomy structure $\mathcal{T}$ as input. Thus, we can also construct training data from $\mathcal{T}$ to fine-tune the LLM. For each seed $s_{l,m} \in \mathcal{S}_l$ $(l \ge 2)$, we train the LLM to pick the correct parent node $\mathsf{Parent}(s_{l,m})$ from $\mathcal{S}_{l-1}$. (Note that to align the INSTRUCTION used in fine-tuning with that in inference, the candidate parent list is $\mathcal{S}_{l-1}$ instead of the entire input taxonomy.) The objective is still maximizing the log-likelihood.

$$\sum_{l=2}^{L}\sum_{m=1}^{M_l} \log \Pr(\,\mathsf{Parent}(s_{l,m}) \mid \text{INSTRUCTION}, \text{INPUT}, \text{RESPONSE}\,). \quad (4)$$

After the LLM is fine-tuned, we apply it to find both "siblings"/ "cousins" and "parents" to complete the seed-guided taxonomy construction task.

A Unified Pre-training Framework

With the above instructions, one can directly prompt/tune an LLM to perform each task separately. However, task-specific training data may be too scarce for the model to acquire sufficient knowledge and skills for finding siblings and parents. For example, the input taxonomy of the seed-guided

taxonomy construction task typically contains about 10 entities (Shen et al. 2018a). To bridge this gap, we propose to first continually pre-train a general-purpose LLM (e.g., Llama-3 8B) on an existing large taxonomy with the aforementioned instructions, expecting the knowledge and skills it learns from pre-training data can be transferred to the three downstream tasks.

Pre-training Data. To largely avoid overlap between pre-training data and evaluation benchmarks in downstream tasks (e.g., Wiki (Ling and Weld 2012), Environment (Bordea, Lefever, and Buitelaar 2016), and DBLP (Shen et al. 2018a)), we adopt only one existing large-scale taxonomy for pre-training: Comparative Toxicogenomics Database (CTD) (Davis et al. 2022), where we take its MEDIC disease vocabulary.

Pre-training Tasks. The pre-training tasks are parent finding and sibling finding, with the parent finding task including both the setting of finding a parent for a single entity and for a list of entities. Given a set of sibling entities $\mathcal{S} = \{s_1, s_2, ..., s_{|\mathcal{S}|}\}$ and their parent $\mathsf{Parent}(\mathcal{S})$, we randomly pick M of them as seeds, where M is very small (e.g., $M = 4$). For ease of notation, we denote the seeds as $s_1, s_2, ..., s_M$. The pre-training objective of sibling finding is to generate $s_{M+1}, ..., s_{|\mathcal{S}|}$ from the seeds:

$$\log \Pr(\ s_{M+1}, ..., s_{|\mathcal{S}|} \mid \textsc{Instruction},\ \textsc{Input},\ \textsc{Response}\ ), \tag{5}$$

where the (INSTRUCTION, INPUT, RESPONSE) schema follows the Entity Set Expansion's sibling-finding template. The pre-training objective of parent finding is to generate $\mathsf{Parent}(\mathcal{S})$ for each individual seed s_i $(1 \leq i \leq M)$ as well as for the entire set of seeds $\{s_1, s_2, ..., s_M\}$:

$$\log \Pr(\ \mathsf{Parent}(\mathcal{S}) \mid \textsc{Instruction},\ \textsc{Input},\ \textsc{Response}\ ), \tag{6}$$

where the (INSTRUCTION, INPUT, RESPONSE) schema follows the Taxonomy Expansion's parent-finding template.

Intuitively, the two pre-training tasks can mutually benefit each other because accurately predicting the siblings $s_{M+1}, ..., s_{|\mathcal{S}|}$ of $s_1, s_2, ..., s_M$ helps inferring the parent $\mathsf{Parent}(\mathcal{S})$ of $s_1, s_2, ..., s_M$, and vice versa.

Experiments

We now demonstrate the effectiveness of TAXOINSTRUCT in all three tasks by comparing it with previous competitive methods on benchmark datasets. We have also included detailed descriptions of metrics and baselines in the Appendix.

Entity Set Expansion

Datasets Following previous studies (Shen et al. 2017; Yan et al. 2019; Zhang et al. 2020), we use two benchmark datasets, **APR** and **Wiki**, to evaluate entity set expansion algorithms. The APR dataset has 15 testing queries related to news articles published by Associated Press and Reuters in 2015; the Wiki dataset contains 40 testing queries related to a subset of English Wikipedia articles.

Baselines We compare TAXOINSTRUCT with the following methods: **(1) EgoSet** (Rong et al. 2016), **(2) SetExpan** (Shen et al. 2017), **(3) SetExpander** (Mamou et al. 2018a), **(4) CaSE** (Yu et al. 2019), **(5) SetCoExpan** (Huang et al. 2020), **(6) CGExpan** (Zhang et al. 2020), **(7) SynSetExpan** (Shen et al. 2020a), **(8) ProbExpan** (Li et al. 2022). representation of entities for entity set expansion. Besides, since TAXOINSTRUCT is pre-trained on both the parent-finding and sibling-finding tasks, to show that the former skill can benefit the latter one, we examine an ablation of TAXOINSTRUCT, **NoParentPretrain**, that is only pre-trained for finding siblings in entity set expansion.

Table 1: Performance of compared methods in the entity set expansion task. Bold: the best score. *: TAXOINSTRUCT is significantly better than this method with p-value < 0.05. †, ‡, and $^\triangleright$: the scores of this method are reported in (Zhang et al. 2020), (Huang et al. 2020), and (Li et al. 2022), respectively.

Method	APR		Wiki	
	MAP@10	MAP@20	MAP@10	MAP@20
EgoSet †	0.758*	0.710*	0.904*	0.877*
SetExpan †	0.789*	0.763*	0.944*	0.921*
SetExpander †	0.287*	0.208*	0.499*	0.439*
CaSE †	0.619*	0.494*	0.897*	0.806*
SetCoExpan ‡	0.933*	0.915*	0.976*	0.964*
CGExpan †	0.992	0.990*	0.995	0.978*
SynSetExpan $^\triangleright$	0.985*	0.990*	0.991*	0.978*
ProbExpan $^\triangleright$	0.993	0.990*	0.995	0.982
TAXOINSTRUCT	**0.9956**	**0.9928**	**0.9957**	**0.9875**
NoParentPretrain	0.9867*	0.9689*	0.9746*	0.9720*

Evaluation Metric Following previous studies (Shen et al. 2017; Zhang et al. 2020), we adopt the Mean Average Precision (**MAP@**k) as the evaluation metric.

Experimental Results Table 1 shows the MAP@10 and 20 scores of compared methods in entity set expansion. We run TAXOINSTRUCT multiple times and report the average performance. To show statistical significance, we conduct a two-tailed Z-test to compare TAXOINSTRUCT with each baseline, and the significance level is marked in Table 1. We can observe that: (1) TAXOINSTRUCT consistently outperforms all baselines, including those empowered by language model probing (e.g., CGExpan and ProbExpan). In most cases, the advantage of TAXOINSTRUCT is statistically significant. (2) TAXOINSTRUCT performs significantly better than NoParentPretrain. This implies that even in the entity set expansion task, where finding siblings is the primarily required skill, pre-training TAXOINSTRUCT to find parents still effectively boosts the performance. This finding validates our motivation for pre-training a unified model to jointly solve different but related tasks.

Taxonomy Expansion

Datasets Following (Jiang et al. 2023b), we use two scientific domain datasets, **Environment** and **Science**, from the shared task in SemEval 2016 (Bordea, Lefever, and Buitelaar 2016). The entities (both existing ones in the input taxonomy and new ones to be inserted) are scientific concepts related to environment and general science, respectively.

Baselines We compare TAXOINSTRUCT with the following methods: **TAXI** (Panchenko et al. 2016), **HypeNET** (Shwartz, Goldberg, and Dagan 2016),

Table 2: Performance of compared methods in the taxonomy expansion task. Bold and *: the same meaning as in Table 1. †, ‡, and ▷: the scores of this method are reported in (Jiang et al. 2023b), (Zeng et al. 2021), and (Liu et al. 2021), respectively.

Method	Environment		Science	
	Acc	Wu&P	Acc	Wu&P
TAXI †	0.167*	0.447*	0.130*	0.329*
HypeNET †	0.167*	0.558*	0.154*	0.507*
BERT+MLP †	0.111*	0.479*	0.115*	0.436*
TaxoExpan †	0.111*	0.548*	0.278*	0.576*
Arborist ‡	0.4615*	–	0.4193*	–
Graph2Taxo ‡	0.2105*	–	0.2619*	–
STEAM †	0.361*	0.696*	0.365*	0.682*
TMN ‡	0.3793*	–	0.3415*	–
TEMP ▷	0.492*	0.777*	0.578*	0.853
GenTaxo ‡	0.4828*	–	0.3878*	–
BoxTaxo †	0.381*	0.754*	0.318*	0.647*
TAXOINSTRUCT	**0.5115**	**0.8300**	**0.6165**	0.8480
NoSiblingPretrain	0.4616*	0.7911*	0.5953*	**0.8559**

BERT+MLP (Devlin et al. 2019), **TaxoExpan** (Shen et al. 2020b), **Arborist** (Manzoor et al. 2020), **Graph2Taxo** (Shang et al. 2020), **STEAM** (Yu et al. 2020), **TMN** (Zhang et al. 2021), **TEMP** (Liu et al. 2021), **GenTaxo** (Zeng et al. 2021), **BoxTaxo** (Jiang et al. 2023b).

We also consider an ablation version of TAXOINSTRUCT, **NoSiblingPretrain**, for the taxonomy expansion task, which mainly relies on the parent-finding skill.

Evaluation Metrics We adopt **Accuracy (Acc) Wu & Palmer Similarity (Wu&P) (Wu and Palmer 1994)** as the evaluation metrics for the taxonomy expansion task.

Previous studies (Yu et al. 2020; Zeng et al. 2021; Jiang et al. 2023b) also consider the mean reciprocal rank (MRR) as an evaluation metric. However, it requires a model to rank all nodes in the taxonomy according to their likelihood of being the parent, which is not applicable to TAXOINSTRUCT that generates only one predicted parent entity.

Experimental Results Table 2 shows the performance of compared methods in taxonomy expansion. We can see that: (1) TAXOINSTRUCT significantly outperforms the baselines in almost all cases. The only exception is that TEMP has a higher Wu&P score on the Science dataset. Besides TEMP, GenTaxo is a competitive baseline which adopts a generative paradigm for taxonomy expansion. However, unlike TAXOINSTRUCT that exploits the power of LLMs to fully unleash the strengths of the generative paradigm, GenTaxo only considers the Gated Recurrent Unit (GRU) architecture, leading to suboptimal performance. (2) TAXOINSTRUCT achieves higher metrics than NoSiblingPretrain in most columns, indicating that even in the taxonomy expansion task, where finding parents is the primarily demanded skill, pre-training our model to accurately find siblings is still helpful. Combining this observation with the one from the ablation analysis in Entity Set Expansion's experiment, we conclude that sibling-finding and parent-finding skills can *mutually* benefit each other.

Table 3: Performance of compared methods in the seed-guided taxonomy construction task. Bold and *: the same meaning as in Table 1.

Method	DBLP		PubMed-CVD	
	Sibling nDCG@50	Parent nDCG@50	Sibling nDCG@50	Parent nDCG@50
HSetExpan	0.881*	0.827*	0.652*	0.509*
NoREPEL	0.883*	0.815*	0.671*	0.622*
NoGTO	0.953*	0.886*	0.740*	0.643*
HiExpan	0.952*	0.905	0.737*	0.713*
TAXOINSTRUCT	**0.982**	**0.921**	**0.922**	**0.8034**
NoParentPretrain	0.967*	0.784*	0.892*	0.7864
NoSiblingPretrain	0.943*	0.911	0.793*	0.6838*

Seed-Guided Taxonomy Construction

Datasets We adopt two scientific domain datasets, **DBLP** and **PubMed-CVD**, introduced in (Shen et al. 2018a). The seeds we use are the same as those in (Shen et al. 2018a). Both datasets have a two-layer input taxonomy. For PubMed-CVD, there are 3 seeds at the top layer (i.e., *Cardiovascular Abnormalities*, *Vascular Diseases*, and *Heart Disease*) and 10 seeds at the bottom layer. For DBLP, there are 5 seeds at the top layer (i.e., *Machine Learning*, *Data Mining*, *Natural Language Processing*, *Information Retrieval*, and *Wireless Networks*) and 11 seeds at the bottom layer.

Baselines We compare TAXOINSTRUCT with the following methods: **HSetExpan** (Shen et al. 2017), **HiExpan** (Shen et al. 2018a). We also add two ablation versions of HiExpan: **HiExpan-NoREPEL** (Shen et al. 2018a), **HiExpan-NoGTO** (Shen et al. 2018a)

Shen et al. (2018a) have released the output taxonomies[3] of the four baselines above on DBLP and PubMed-CVD, which we use for evaluation. In addition to these four baselines, following our practice in the previous two tasks, we consider two ablation versions, **NoParentPretrain** and **NoSiblingPretrain**.

Evaluation Metrics At the top layer, most baselines and our TAXOINSTRUCT model achieve near perfect accuracy. Therefore, our evaluation metrics focus on the more challenging bottom layer. We use **Sibling nDCG@**k to evaluates the accuracy of the sibling-finding step and **Parent nDCG@**k for the accuracy of the parent-finding step. Formally, given the bottom-layer seeds $\mathcal{S}_2 = \{s_{2,1}, ..., s_{2,M}\}$, we examine the top-$k$ expanded bottom-layer entities $\mathcal{S}_2^+ = \{s_{2,M+1}, ..., s_{2,M+k}\}$.

Experimental Results Table 3 demonstrates the Parent and Sibling nDCG@50 scores of compared methods in seed-guided taxonomy construction. We find that: (1) TAXOINSTRUCT performs evidently the best in both sibling-finding and parent-finding steps on both datasets. Note that expanding correct sibling terms which are relevant to the taxonomy serves as the prerequisite of finding correct parents for the expanded terms. If an expanded sibling is wrong (i.e., it should not appear at this layer or even in the taxonomy),

[3] http://bit.ly/2Jbilte

then it is impossible to predict its correct parent. This explains why a Sibling nDCG@50 score is always higher than the corresponding Parent nDCG@50 score. (2) TAXOINSTRUCT consistently beats the two ablation versions, which is intuitive because seed-guided taxonomy construction requires the collaboration of the two skills. NoSiblingPretrain has a lower Sibling nDCG@50 score than NoParentPretrain on both datasets (because the former is not pre-trained for sibling finding). However, on DBLP, NoSiblingPretrain achieves a much smaller performance drop between Sibling nDCG@50 and Parent nDCG@50 scores than NoParentPretrain due to its strength in parent finding.

Related Work

Entity Set Expansion. EgoSet (Rong et al. 2016) is a pioneering work that utilizes skip-grams and word2vec embeddings (Mikolov et al. 2013) to perform entity set expansion. Following this idea, SetExpan (Shen et al. 2017) proposes an iterative bootstrapping framework to select indicative skip-grams and gradually expand the entity set; SetExpander (Mamou et al. 2018b) and CaSE (Yu et al. 2019) leverage distributional similarity obtained from context-free embedding learning to rank candidate entities according to the seeds; SetCoExpan (Huang et al. 2020) generates auxiliary sets as negative sets and then expand multiple sets simultaneously to extract discriminative features. With the emergence of pre-trained language models such as BERT (Devlin et al. 2019), related studies propose to replace context-free embeddings with contextualized representations. For example, CGExpan (Zhang et al. 2020) uses BERT to automatically generate class names as a stronger signal to prevent semantic drifting; ProbExpan (Li et al. 2022) devises an entity-level masked language model with contrastive learning to refine the representation of entities; GAPA (Li et al. 2023a) proposes a context pattern generation module that uses autoregressive language models (e.g., GPT-2 (Radford et al. 2019)). However, all aforementioned approaches do not explore the power of LLMs with billions of parameters and the ability to follow instructions, while TAXOINSTRUCT extensively exploits the effectiveness of LLMs in entity set expansion.

Taxonomy Expansion. Earlier, lexical patterns (Panchenko et al. 2016) and distributional word representations (Shwartz, Goldberg, and Dagan 2016) are used to infer the hypernym-hyponym relationship. Later, many attempts have focused on exploiting the graph structure in the input taxonomy to enhance the performance of taxonomy expansion. For example, TaxoExpan (Shen et al. 2020b) and STEAM (Yu et al. 2020) propose to encode local ego-graphs and mini-paths, respectively, corresponding to each entity in the taxonomy; Arborist (Manzoor et al. 2020) considers heterogeneous edge semantics and optimizes the shortest-path distance between predicted and actual parents; GraphTaxo (Shang et al. 2020) proposes a directed acyclic graph generation method for effective domain transfer; TMN (Zhang et al. 2021) examines both candidate parents and candidate children of the inserted query node via a triplet matching network; Most recently, GenTaxo (Zeng et al. 2021) presents a GRU-based decoder to generate concepts for taxonomy completion; TaxoPrompt (Xu et al. 2022) and TacoPrompt (Xu et al. 2023) adopt prompt tuning on BERT-based encoder model to generate contextualized representations of the global taxonomy structure; BoxTaxo (Jiang et al. 2023b) uses box embeddings to replace single-vector embeddings to better capture the hierarchical structure of concepts. Introducing a more challenging version of taxonomy expansion, Shen et al. (2018a) study seed-guided taxonomy construction which requires the initial step of extracting new entities from text corpora given a small set of seeds before performing taxonomy expansion. Different from previous approaches that utilize context-free embeddings, graph neural networks, and BERT-based language models, our TAXOINSTRUCT model unleashes the power of LLMs such as Llama-2. Moreover, TAXOINSTRUCT is a unified framework aiming to jointly solve entity set expansion, taxonomy expansion, and seed-guided taxonomy construction rather than any of them alone.

Structure-Aware Instruction Tuning. Inspired by the great success of LLMs in dealing with text data, there has been increasing attention on utilizing LLMs to learn from (text-rich) structured data (Jin et al. 2023; Li et al. 2023b; Chen et al. 2024). Previous approaches have been using instruction tuning to guide LLMs to acquire structural information. For instance, Graph-ToolFormer (Zhang 2023) empowers LLMs with graph reasoning abilities via prompt-augmented by ChatGPT; Wang et al. (2023a) propose Build-a-Graph Prompting and Algorithmic Prompting techniques to enhance LLMs in solving graph problems such as shortest paths and maximum flows; Guo, Du, and Liu (2023) conduct an empirical benchmark study on LLMs' ability to understand graph data by using formal language to describe graphs; InstructGLM (Ye et al. 2023) demonstrates that LLMs fine-tuned on node classification and link prediction tasks (with proper designs of instructions) can outperform competitive graph neural network baselines; Zhang et al. (2023) put entity triplets into an instruction template as the LLM's input to perform knowledge graph completion. Different from these studies that focus on graph structures (e.g., academic and e-commerce networks), our work specifically explores how taxonomy structures can guide the instruction tuning process to unleash LLMs' potential to solve entity enrichment tasks in a unified way.

Conclusions and Future Work

In this paper, we present TAXOINSTRUCT, a unified framework to jointly solve entity set expansion, taxonomy expansion, and seed-guided taxonomy construction. We propose a taxonomy-guided instruction tuning technique that effectively exploits the existing large-scale taxonomy to teach large language models the commonality of the three tasks (i.e., the skills of sibling finding and parent finding). We conduct extensive experiments on widely used benchmarks in all three tasks, which demonstrate the superiority of TAXOINSTRUCT over competitive task-specific baselines.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bordea, G.; Lefever, E.; and Buitelaar, P. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *SemEval'16*, 1081–1091.

Chen, Z.; Mao, H.; Li, H.; Jin, W.; Wen, H.; Wei, X.; Wang, S.; Yin, D.; Fan, W.; Liu, H.; et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2): 42–61.

Christmann, P.; Saha Roy, R.; and Weikum, G. 2022. Conversational question answering on heterogeneous sources. In *SIGIR'22*, 144–154.

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. S. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *ACL'20*, 2270–2282.

Coletti, M. H.; and Bleich, H. L. 2001. Medical subject headings used to search the biomedical literature. *JAMIA*, 8(4): 317–323.

Davis, A. P.; Wiegers, T. C.; Johnson, R. J.; Sciaky, D.; Wiegers, J.; and Mattingly, C. J. 2022. Comparative Toxicogenomics database (CTD): update 2023. *Nucleic Acids Research*, 51(D1): D1257–D1262.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT'19*, 4171–4186.

Guo, J.; Du, L.; and Liu, H. 2023. GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking. *arXiv preprint arXiv:2305.15066*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, J.; Xie, Y.; Meng, Y.; Shen, J.; Zhang, Y.; and Han, J. 2020. Guiding corpus-based set expansion by auxiliary sets generation and co-expansion. In *WWW'20*, 2188–2198.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023a. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jiang, M.; Song, X.; Zhang, J.; and Han, J. 2022. Taxoenrich: Self-supervised taxonomy completion via structure-semantic representations. In *WWW'22*, 925–934.

Jiang, S.; Yao, Q.; Wang, Q.; and Sun, Y. 2023b. A Single Vector Is Not Enough: Taxonomy Expansion via Box Embeddings. In *WWW'23*, 2467–2476.

Jin, B.; Liu, G.; Han, C.; Jiang, M.; Ji, H.; and Han, J. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.

Li, J.; Zhao, W. X.; Wei, Z.; Yuan, N. J.; and Wen, J.-R. 2021. Knowledge-based review generation by coherence enhanced text planning. In *SIGIR'21*, 183–192.

Li, Y.; Huang, S.; Zhang, X.; Zhou, Q.; Li, Y.; Liu, R.; Cao, Y.; Zheng, H.-T.; and Shen, Y. 2023a. Automatic Context Pattern Generation for Entity Set Expansion. *IEEE TKDE*, 35(12): 12458–12469.

Li, Y.; Li, Y.; He, Y.; Yu, T.; Shen, Y.; and Zheng, H.-T. 2022. Contrastive learning with hard negative entities for entity set expansion. In *SIGIR'22*, 1077–1086.

Li, Y.; Li, Z.; Wang, P.; Li, J.; Sun, X.; Cheng, H.; and Yu, J. X. 2023b. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*.

Ling, X.; and Weld, D. 2012. Fine-grained entity recognition. In *AAAI'12*, 94–100.

Liu, Z.; Xu, H.; Wen, Y.; Jiang, N.; Wu, H.; and Yuan, X. 2021. TEMP: taxonomy expansion with dynamic margin loss through taxonomy-paths. In *EMNLP'21*, 3854–3863.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. In *ICLR'19*.

Mamou, J.; Pereg, O.; Wasserblat, M.; Dagan, I.; Goldberg, Y.; Eirew, A.; Green, Y.; Guskin, S.; Izsak, P.; and Korat, D. 2018a. Setexpander: End-to-end term set expansion based on multi-context term embeddings. In *COLING'18 System Demonstrations*, 58–62.

Mamou, J.; Pereg, O.; Wasserblat, M.; Eirew, A.; Green, Y.; Guskin, S.; Izsak, P.; and Korat, D. 2018b. Term Set Expansion based NLP Architect by Intel AI Lab. In *EMNLP'18 System Demonstrations*, 19–24.

Manzoor, E.; Li, R.; Shrouty, D.; and Leskovec, J. 2020. Expanding taxonomies with implicit edge semantics. In *WWW'20*, 2044–2054.

Mao, Y.; Zhao, T.; Kan, A.; Zhang, C.; Dong, X. L.; Faloutsos, C.; and Han, J. 2020. Octet: Online catalog taxonomy enrichment with self-supervision. In *KDD'20*, 2247–2257.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS'13*, 3111–3119.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS'22*, 27730–27744.

Panchenko, A.; Faralli, S.; Ruppert, E.; Remus, S.; Naets, H.; Fairon, C.; Ponzetto, S. P.; and Biemann, C. 2016. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *SemEval'16*, 1320–1327.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Rong, X.; Chen, Z.; Mei, Q.; and Adar, E. 2016. Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In *WSDM'16*, 645–654.

Shang, C.; Dash, S.; Chowdhury, M. F. M.; Mihindukulasooriya, N.; and Gliozzo, A. 2020. Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer. In *ACL'20*, 2198–2208.

Shen, J.; Qiu, W.; Shang, J.; Vanni, M.; Ren, X.; and Han, J. 2020a. SynSetExpan: An iterative framework for joint entity set expansion and synonym discovery. In *EMNLP'20*, 8292–8307.

Shen, J.; Shen, Z.; Xiong, C.; Wang, C.; Wang, K.; and Han, J. 2020b. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *WWW'20*, 486–497.

Shen, J.; Wu, Z.; Lei, D.; Shang, J.; Ren, X.; and Han, J. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *ECML-PKDD'17*, 288–304.

Shen, J.; Wu, Z.; Lei, D.; Zhang, C.; Ren, X.; Vanni, M. T.; Sadler, B. M.; and Han, J. 2018a. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *KDD'18*, 2180–2189.

Shen, J.; Xiao, J.; He, X.; Shang, J.; Sinha, S.; and Han, J. 2018b. Entity set search of scientific literature: An unsupervised ranking approach. In *SIGIR'18*, 565–574.

Shwartz, V.; Goldberg, Y.; and Dagan, I. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *ACL'16*, 2389–2398.

Team, G.; Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, H.; Feng, S.; He, T.; Tan, Z.; Han, X.; and Tsvetkov, Y. 2023a. Can language models solve graph problems in natural language? In *NeurIPS'23*.

Wang, R. C.; and Cohen, W. W. 2007. Language-independent set expansion of named entities using the web. In *ICDM'07*, 342–350.

Wang, S.; Zhao, R.; Chen, X.; Zheng, Y.; and Liu, B. 2021. Enquire one's parent and child before decision: Fully exploit hierarchical structure for self-supervised taxonomy expansion. In *WWW'21*, 3291–3304.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023b. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *ACL'23*, 13484–13508.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models are Zero-Shot Learners. In *ICLR'22*.

Wu, Z.; and Palmer, M. 1994. Verbs semantics and lexical selection. In *ACL'94*, 133–138.

Xu, H.; Chen, Y.; Liu, Z.; Wen, Y.; and Yuan, X. 2022. TaxoPrompt: A Prompt-based Generation Method with Taxonomic Context for Self-Supervised Taxonomy Expansion. In *IJCAI'22*, 4432–4438.

Xu, H.; Liu, C.; Niu, Y.; Chen, Y.; Cai, X.; Wen, Y.; and Yuan, X. 2023. TacoPrompt: A Collaborative Multi-Task Prompt Learning Method for Self-Supervised Taxonomy Completion. In *EMNLP'23*, 15804–15817.

Yan, L.; Han, X.; Sun, L.; and He, B. 2019. Learning to bootstrap for entity set expansion. In *EMNLP'19*, 292–301.

Ye, R.; Zhang, C.; Wang, R.; Xu, S.; and Zhang, Y. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*.

Yu, P.; Huang, Z.; Rahimi, R.; and Allan, J. 2019. Corpus-based set expansion with lexical features and distributed representations. In *SIGIR'19*, 1153–1156.

Yu, Y.; Li, Y.; Shen, J.; Feng, H.; Sun, J.; and Zhang, C. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *KDD'20*, 1026–1035.

Zeng, Q.; Lin, J.; Yu, W.; Cleland-Huang, J.; and Jiang, M. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *KDD'21*, 2104–2113.

Zhang, J. 2023. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*.

Zhang, J.; Song, X.; Zeng, Y.; Chen, J.; Shen, J.; Mao, Y.; and Li, L. 2021. Taxonomy completion via triplet matching network. In *AAAI'21*, 4662–4670.

Zhang, Y.; Chen, Z.; Zhang, W.; and Chen, H. 2023. Making Large Language Models Perform Better in Knowledge Graph Completion. *arXiv preprint arXiv:2310.06671*.

Zhang, Y.; Shen, J.; Shang, J.; and Han, J. 2020. Empower Entity Set Expansion via Language Model Probing. In *ACL'20*, 8151–8160.

Appendix

1. Implementation Detail

We use Llama-3 8B as our initial checkpoint and train it with Low-Rank Adaptation (LoRA) (Hu et al. 2021). The model is pre-trained for 10 epochs, which takes about 1.5 hours on one NIVIDIA RTX A6000 GPU. The batch size is 64 for both continual pre-training and task-specific fine-tuning. The optimizer is AdamW (Loshchilov and Hutter 2017). When performing downstream tasks, for entity set expansion and taxonomy expansion, we adopt SPECTER (Cohan et al. 2020) as the moderate-size auxiliary PLM.

2. Evaluation Metrics

2.1 Entity Set Expansion We use AP@k and MAP@k as evaluation metrics for Entity Set Expansion. Formally, given a set of seeds $\mathcal{S} = \{s_1, ..., s_M\}$ and the top-k expanded entities $\mathcal{S}^+ = \{s_{M+1}, ..., s_{M+k}\}$, the average precision AP@k is defined as

$$\text{AP@}k(\mathcal{S}, \mathcal{S}^+) = \frac{1}{k} \sum_{i:1\le i\le k \text{ and } s_{M+i}\sim\mathcal{S}} \frac{\sum_{j=1}^{i} \mathbb{I}(s_{M+j} \sim \mathcal{S})}{i}. \tag{7}$$

Here, $s_{M+j} \sim \mathcal{S}$ denotes that the expanded entity s_{M+j} and the seed entities in $\mathcal{S}$ belong to the same semantic class; $\mathbb{I}(\cdot)$ is the indicator function. Since there are multiple testing queries (i.e., multiple sets of seeds) $\mathcal{S}_1, ..., \mathcal{S}_C$ and their corresponding expansion results $\mathcal{S}_1^+, ..., \mathcal{S}_C^+$, MAP@$k$ is defined as

$$\text{MAP@}k = \frac{1}{C} \sum_{i=1}^{C} \text{AP@}k(\mathcal{S}_i, \mathcal{S}_i^+). \tag{8}$$

2.2 Taxonomy Expansion. We use **Accuracy (Acc)** Wu&P as evaluation metrics for Taxonomy Expansion. **Accuracy (Acc)** is the exact match accuracy of the predicted parent node of each testing entity. Formally, assume the testing set has C samples $x_1, ..., x_C$, and their ground-truth parents in the input taxonomy are $y_1, ..., y_C$, respectively. Then the accuracy of the learned parent-child relationship $\mathsf{Parent}^+(\cdot)$ is defined as

$$\text{Acc} = \frac{1}{C} \sum_{i=1}^{C} \mathbb{I}(\mathsf{Parent}^+(x_i) = y_i). \tag{9}$$

Wu & Palmer Similarity (Wu&P) (Wu and Palmer 1994) calculates the similarity between the predicted parent and the ground-truth parent based on their distance in the taxonomy.

$$\text{Wu\&P} = \frac{1}{C} \sum_{i=1}^{C} \frac{2 \times \text{depth}(\text{LCP}(\mathsf{Parent}^+(x_i), y_i))}{\text{depth}(\mathsf{Parent}^+(x_i)) + \text{depth}(y_i)}, \tag{10}$$

where $\text{LCP}(\cdot, \cdot)$ is the lowest common ancestor of two nodes, and $\text{depth}(\cdot)$ denotes the depth of a node in the taxonomy.

2.3 Seed-Guided Taxonomy Construction. We evaluate the performance of Seed-Guided Taxonomy Construction using **Sibling nDCG@k** and **Parent nDCG@k**. **Sibling nDCG@k** evaluates the accuracy of the sibling-finding step (i.e., whether $s_{2,M+i}$ and $\mathcal{S}_2$ belong to the same semantic class).

$$\text{Sibling nDCG@}k = \frac{\sum_{i=1}^{k} \frac{\mathbb{I}(s_{2,M+i}\sim\mathcal{S}_2)}{\log_2(i+1)}}{\sum_{i=1}^{k} \frac{1}{\log_2(i+1)}}. \tag{11}$$

Parent nDCG@k evaluates the accuracy of the parent-finding step. For each expanded bottom-layer entity $s_{2,M+i}$, let $s_{1,p(i)}$ denote its ground-truth parent at the top layer. Then, this metric can be defined as

$$\text{Parent nDCG@}k = \frac{\sum_{i=1}^{k} \frac{\mathbb{I}(\mathsf{Parent}^+(s_{2,M+i})=s_{1,p(i)})}{\log_2(i+1)}}{\sum_{i=1}^{k} \frac{1}{\log_2(i+1)}}. \tag{12}$$

Table 4: Performance of TAXOINSTRUCT with different LLM backbones. For the seed-guided taxonomy construction task (i.e., DBLP and PubMed-CVD), we show Sibling nDCG@50; for the taxonomy expansion task (i.e., Environment and Science), we show Wu&P.

Method	DBLP	PubMed-CVD	Environment	Science
Strongest Baseline	0.9527	0.7395	0.777	0.853
TAXOINSTRUCT				
Llama-3 8B	**0.9817**	**0.9220**	**0.8300**	0.8480
Llama-2-chat 7B	0.9713	0.8923	0.7739	0.7370
Mistral 7B	0.9635	0.9162	0.7552	0.8437
Gemma 7B	0.9685	0.8627	0.7893	**0.8713**

3. Effect of the LLM Backbone

Although we adopt Llama-3 8B as the backbone of TAXOINSTRUCT in our experiments, we need to emphasize that TAXOINSTRUCT is a generic framework that can be instantiated by various off-the-shelf generative LLMs. To show the generalizability of TAXOINSTRUCT, we examine the performance of TAXOINSTRUCT when Llama-2-chat 7B (Touvron et al. 2023)[4], Mistral 7B (Jiang et al. 2023a)[5], and Gemma 7B (Team et al. 2024)[6] are plugged in.

Table 4 demonstrates the performance of TAXOINSTRUCT with different LLM backbones. Due to space limit, we only show 4 datasets (out of the 6 benchmarks we used in previous experiments) and 1 metric for each dataset. We can see that: (1) On DBLP and PubMed-CVD, all the variants of TAXOINSTRUCT beat the strongest baseline, no matter which LLM backbone is plugged in. (2) On the Environment dataset, both Llama-3 8B and Gemma 7B can make our framework more powerful than the best-performing baseline. (3) On the Science dataset, even our default choice Llama-3 8B does not perform the best in Table 3, when TAXOINSTRUCT is instantiated by Gemma 7B, it can beat the state of the art. To summarize, the effectiveness of TAXOINSTRUCT is built upon the power of our proposed framework and LLMs *in general*, rather than a specific choice of Llama-3 8B.

4. Scope and Limitations.

This work is a pioneering attempt to solve the three representative expansion tasks in a unified way. Our major focus

[4]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[5]https://huggingface.co/mistralai/Mistral-7B-v0.1

[6]https://huggingface.co/google/gemma-7b

is to verify the universal validity of multi-task LLM instruction tuning in all three tasks. Therefore, we keep our framework as simple as we can, without utilizing complicated signals such as paths (Liu et al. 2021; Jiang et al. 2022) and local graphs (Mao et al. 2020; Wang et al. 2021). We are aware that incorporating these signals into our instructions may further improve the performance of TAXOINSTRUCT, but that is beyond the scope of this paper, and we leave it for future work.