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

Anonymous ACL submission

Abstract

Entity set expansion, taxonomy expansion, and seed-guided taxonomy construction are three representative tasks for automatically enriching an existing taxonomy with emerging concepts. Previous studies have treated them as separate tasks, leading to techniques that are specialized for one task but lack generalizability and a holistic perspective. In this paper, we propose a unified solution to address all three tasks. Specifically, we identify two fundamental skills facilitating the three tasks: finding "siblings" and finding "parents". To this end, we introduce a taxonomy-guided instruction tuning framework that trains a large language model to generate siblings and parents for query entities, where the joint pre-training process enables mutual reinforcement of these two skills. Extensive experiments on multiple benchmark datasets validate the effectiveness of our proposed TAXOINSTRUCT framework, demonstrating its superiority over task-specific baselines across all three tasks.

1 Introduction

Entities are fundamental to natural language processing. To better capture their semantics, taxonomies are constructed across various domains, including science (Shen et al., 2018b), e-commerce (Mao et al., 2020), and social media (Gonçalves et al., 2019), to characterize the parent-child relationship between entities. While taxonomies are often initially curated by domain experts, the continuous emergence of new concepts necessitates automatic expansion to maintain their freshness and completeness. To this end, previous studies have explored three key tasks for integrating 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 specific semantic class, the goal is to identify more entities within the same class. For example, given the seed entities {*Database*, *Information Retrieval*, *Operating System*}, an entity set expansion algorithm should retrieve other computer science subfields such as *Data Mining* and *Human-Computer Interaction*. From a 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 goal of this task is to insert a provided new entity into an existing taxonomy by identifying its most appropriate "**parents**". For instance, consider a taxonomy with 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 place 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 goal is to construct a more comprehensive taxonomy that expands upon the initial structure. For example, if the input consists of *Computer Science*, *Chemistry*, and several of their subfields (e.g., *Data Mining* and *Organic Chemistry*), the expected output should include more scientific fields (e.g., *Mathematics* and *Physics*) and their subfields (e.g., *Database*, *Algebra*, and *Astrophysics*), with explicitly identified parent-child relationships. To approach this problem, we can first discover new entities at each layer and then figure out the parent-child edges between adjacent layers. Essentially, this can be framed as pipelining the steps of finding "**siblings**" and finding "**parents**".

As evident from the discussion above, all three tasks can be cast as finding entities that share a specific type of relationship with the given entities: entity set expansion involves finding "siblings"; taxonomy expansion relies on finding "parents"; seed-guided taxonomy construction integrates both. However, existing studies typically address only

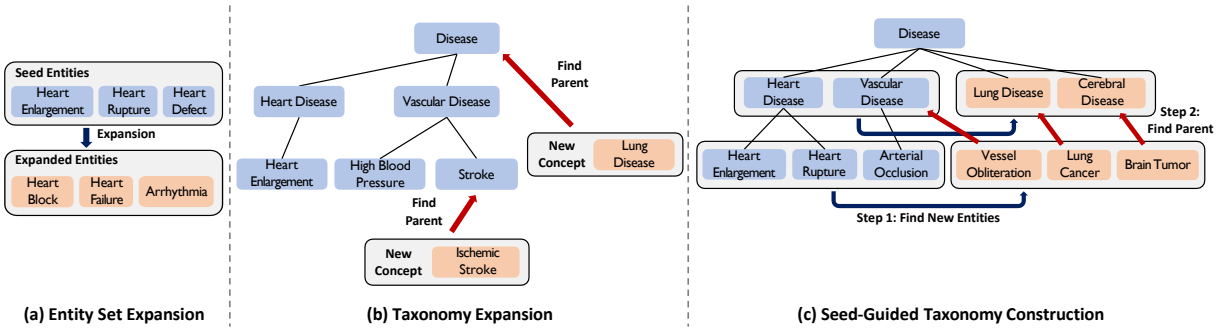

Figure 1: Illustrations of the three tasks.

one of the three tasks, proposing task-specific techniques with little attention to their underlying commonalities. Intuitively, the processes of finding "siblings" and "parents" can reinforce each other. For example, recognizing that *Data Mining* is a sibling of *Database* and *Information Retrieval* can help predict its parent as *Computer Science*, and vice versa. By improving the accuracy of both sibling and parent prediction, we can leverage them as fundamental building blocks to solve all three tasks in a more holistic and unified manner.

Contributions. Building on the insights above, this paper proposes a unified framework to simultaneously address entity set expansion, taxonomy expansion, and seed-guided taxonomy construction. Specifically, we leverage existing taxonomies as rich sources of sibling-sibling and parent-child relationships to pre-train a model for identifying both siblings and parents. This pre-trained model can then be fine-tuned on domain-specific data (e.g., parent-child pairs from the input taxonomy in the taxonomy expansion task) to perform downstream tasks effectively. To implement this framework, we harness the instruction-following capabilities of large language models (LLMs) (Wei et al., 2022a; Ouyang et al., 2022). Our proposed TAXOINSTRUCT framework employs task-specific instructions to train an LLM to generate sibling entities and identify parent entities for one or more query entities. The joint pre-training process enables mutual enhancement between these two skills and benefits overall performance of all three tasks.

To evaluate TAXOINSTRUCT, we conduct extensive experiments on 6 benchmark datasets spanning entity set expansion, taxonomy expansion, and seed-guided taxonomy construction. The results demonstrate that TAXOINSTRUCT, as a unified framework, significantly outperforms strong task-specific baselines across all three tasks. Additionally, we examine the impact of different LLM backbones (Touvron et al., 2023; Jiang et al., 2023a; Team et al., 2024) within TAXOINSTRUCT, showing that its effectiveness is robust and does not depend on a specific LLM choice.

2 Task 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), the entity set expansion task seeks to identify a set of "sibling" entities that belong to the same semantic class as a few example entities (referred to as "seeds"). Formally,

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), taxonomy expansion involves inserting a set of new entities into an existing taxonomy by identifying an appropriate "parent" node in the 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* $\text{PARENT}(\cdot) : \mathcal{S} \rightarrow \mathcal{S} \cup \{s_{\text{ROOT}}\}$*) 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* $\text{PARENT}^+(\cdot) : \mathcal{S} \cup \mathcal{S}^+ \rightarrow \mathcal{S} \cup \{s_{\text{ROOT}}\}$.

Seed-Guided Taxonomy Construction. As shown in Figure 1(c), seed-guided taxonomy construction involves two steps: first, identifying a set of new entities to be added to the taxonomy, and then determining the appropriate parent for each new entity.

Definition 3. *(Seed-Guided Taxonomy Construction) Given a small set of seeds that form a*

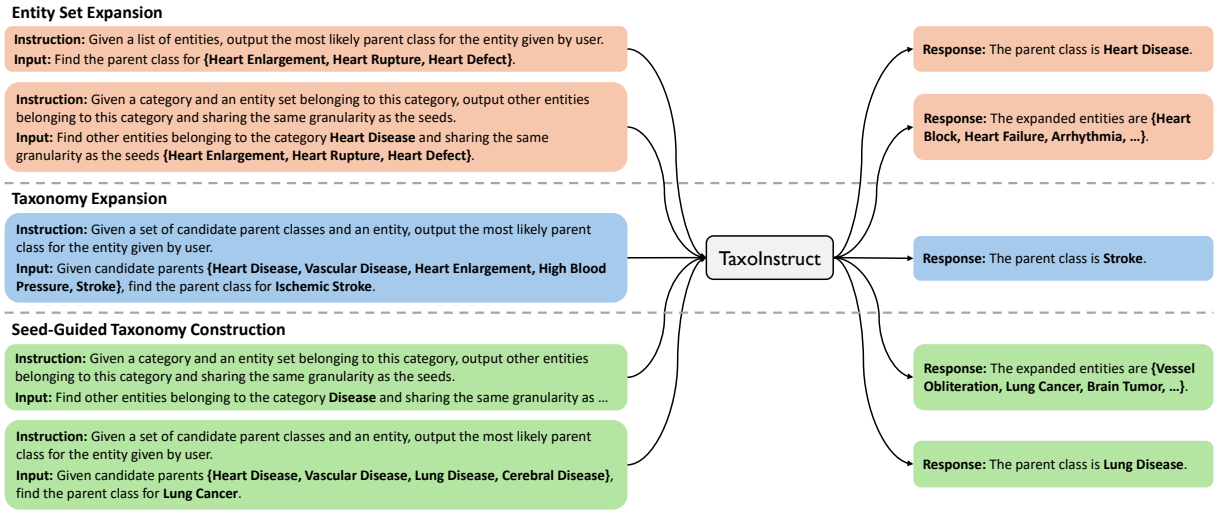

Figure 2: Illustration of the TAXOINSTRUCT framework.

tree structure $\mathcal{T} = (\mathcal{S}_0, \mathcal{S}_1, ..., \mathcal{S}_L)$*, where* $\mathcal{S}_0$ *is* $\{s_{\text{ROOT}}\}$*,* $\mathcal{S}_l$ *(*$1 \leq l \leq L$*) denotes the set of seeds at layer* l*, and the parent-child relationship is characterized by a mapping function* $\text{PARENT}(\cdot) : \mathcal{S}_l \rightarrow \mathcal{S}_{l-1}$*, 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* $\text{PARENT}^+(\cdot) : \mathcal{S}_l \cup \mathcal{S}_l^+ \rightarrow \mathcal{S}_{l-1} \cup \mathcal{S}_{l-1}^+$*).*

3 Model

Inspired by the intuition that entity set expansion, taxonomy expansion, and seed-guided taxonomy construction all rely on two fundamental skills—finding "siblings" and finding "parents"—we aim to train a unified model that simultaneously learns both skills, thereby facilitating all three tasks. To implement this idea, in this section, we propose TAXOINSTRUCT, a unified taxonomy-guided instruction tuning framework.

3.1 Entity Set Expansion

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

Inspired by this, we frame the entity set expansion task (from a taxonomy perspective) as identifying other siblings of the seed entities. We tackle this problem by leveraging the ability of LLMs to follow task-specific instructions (Wei et al., 2022a; Ouyang et al., 2022). Briefly, given a set of INPUT entities $\mathcal{S} = \{s_1, s_2, ..., s_M\}$ that share the same parent node $\text{PARENT}(\mathcal{S})$, we INSTRUCT an LLM (e.g., Llama-3 8B (Dubey et al., 2024)) to generate more children of $\text{PARENT}(\mathcal{S})$ in its RESPONSE.

Nevertheless, the parent entity $\text{PARENT}(\mathcal{S})$ is not available in the standard entity set expansion task (Rong et al., 2016; Shen et al., 2017). 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 $\text{PARENT}(\mathcal{S})$ is then used to guide the expansion process:

> 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* $\text{PARENT}(\mathcal{S})$ *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 en-

tities, 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 pre-trained encoder language model (e.g., BERT (Devlin et al., 2019)) to compute the similarity score between each generated entity $r \in \mathcal{R}$ and $\text{PARENT}(\mathcal{S})$:

$$\text{sim}(r, \text{PARENT}(\mathcal{S})) = \cos\Big(\text{E}(r), \text{E}(\text{PARENT}(\mathcal{S}))\Big), \quad (1)$$

where $\text{E}(\cdot)$ denotes the average output token embedding after feeding the entity name into the pre-trained encoder. All entities in $\mathcal{R}$ are then ranked according to $\text{sim}(\cdot, \text{PARENT}(\mathcal{S}))$. Afterwards, we add the top-ranked entities to the seed entity set $\mathcal{S}$ 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 $\text{sim}(\cdot, \text{PARENT}(\mathcal{S}))$ and obtain a list, $\mathcal{S}^+$, of expanded entities.

3.2 Taxonomy Expansion

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

> 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 a large number of (e.g., more than 10,000) entities (Shen et al., 2020b). 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 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.

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

The retrieved subset $\mathcal{U}_q$ will replace the entire candidate list in the INPUT.

Since the input taxonomy contains a wealth of (parent, child) entity pairs, we leverage this information to fine-tune the LLM, enhancing its understanding of parent-child relationships and domain-specific knowledge. To be specific, given a node s_i in the input taxonomy and its parent $\text{PARENT}(s_i)$, we construct fine-tuning data in two different ways.

First, we take the siblings of $\text{PARENT}(s_i)$ as distractors. In other words, the LLM needs to identify the true parent $\text{PARENT}(s_i)$ from the candidates $\{\text{PARENT}(s_i)\} \cup \text{SIBLING}(\text{PARENT}(s_i))$.

Second, 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 $\text{PARENT}(s_i)$ from the candidates $\{\text{PARENT}(s_i)\} \cup \mathcal{U}_i$.

Filling s_i and the candidates into our instruction template, we fine-tune the LLM to generate $\text{PARENT}(s_i)$.

3.3 Seed-Guided Taxonomy Construction

As shown in Figure 1(c), 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"[1]) and (2) expanding the taxonomy by specifying the proper "parent" for each new entity. Since these two subtasks closely align with entity set expansion and taxonomy expansion, respectively, we can leverage similar instructions as outlined in Sections 3.1 and 3.2.

Finding "Siblings" and "Cousins". Given the input taxonomy $\mathcal{T} = (\mathcal{S}_0, \mathcal{S}_1, ..., \mathcal{S}_L)$ where $\mathcal{S}_0 = \{s_{\text{ROOT}}\}$ and $\mathcal{S}_l = \{s_{l,1}, s_{l,2}, ..., s_{l,M_l}\}$ $(1 \le l \le L)$, we adopt 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* 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 $\text{PARENT}(\mathcal{S}_l)$ into the INPUT to discover not only "siblings" but also "cousins" of $\mathcal{S}_l$. We

[1] In the first step of seed-guided taxonomy construction, the goal is to find entities that share the same semantic granularity as the seeds at each layer. These entities are required only to be descendants of the root node and may not necessarily share the same parent as the seeds. Therefore, this step involves discovering not just "siblings" but also "cousins".

denote the expanded entities at layer l as $\mathcal{S}_l^+ = \{s_{l,M_l+1}, s_{l,M_l+2}, ..., s_{l,M_l+N_l}\}$ $(1 \leq l \leq L)$.

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 $l \geq 2$, 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. Following Section 3.2, for each seed $s_{l,m} \in \mathcal{S}_l$ $(l \geq 2)$, we train the LLM to pick the correct parent node $\text{PARENT}(s_{l,m})$ from $\mathcal{S}_{l-1}$.

3.4 A Unified Pre-training Framework

With the above instructions, an LLM can be directly prompted or fine-tuned to perform each task separately. However, task-specific training data may be too limited for the model to effectively learn the necessary skills for identifying siblings and parents. For instance, the input taxonomy for seed-guided taxonomy construction typically contains about 10 entities only (Shen et al., 2018a). To address this limitation, we propose to first continuously pre-train a general-purpose LLM on a large existing taxonomy using the aforementioned instructions. This pre-training step allows the model to acquire broader knowledge and skills, which can then be transferred to the three tasks, enhancing its performance even with limited task-specific data.

Pre-training Data. To largely avoid overlap between pre-training data and evaluation benchmarks in downstream tasks (e.g., Wikipedia, SemEval, and DBLP), we adopt only one existing large-scale taxonomy for pre-training: Comparative Toxicogenomics Database (CTD) (Davis et al., 2022), where we take its MEDIC taxonomy of disease entities.

Pre-training Tasks. Given a set of sibling entities $\mathcal{S} = \{s_1, s_2, ..., s_{|\mathcal{S}|}\}$ and their parent $\text{PARENT}(\mathcal{S})$ in the taxonomy used for pre-training, we randomly pick M entities from $\mathcal{S}$ as seeds. For ease of notation, we denote the seeds as $s_1, s_2, ..., s_M$.

For the sibling-finding task, the pre-training objective is to generate $s_{M+1}, ..., s_{|\mathcal{S}|}$ from the seeds, where the instruction follows the sibling-finding template in Section 3.1. For the parent-finding task, the pre-training objective is to generate $\text{PARENT}(\mathcal{S})$ for each individual seed s_i $(1 \leq i \leq M)$ as well as for the entire set of seeds $\{s_1, s_2, ..., s_M\}$, where the instruction follows the parent-finding template introduced in Section 3.2. Intuitively, the two pre-training tasks 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 $\text{PARENT}(\mathcal{S})$ of $s_1, s_2, ..., s_M$, and vice versa.

4 Experiments

We now evaluate the effectiveness of TAXOINSTRUCT across all three tasks by comparing it with competitive baselines on benchmark datasets. Details of the baselines and evaluation metrics are provided in Appendices A.1 and A.2, respectively.

4.1 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 two datasets are derived from news articles (published by Associated Press and Reuters) and Wikipedia articles, respectively.

Baselines. The baselines for entity set expansion include **EgoSet** (Rong et al., 2016), **SetExpan** (Shen et al., 2017), **SetExpander** (Mamou et al., 2018), **CaSE** (Yu et al., 2019), **SetCoExpan** (Huang et al., 2020), **CGExpan** (Zhang et al., 2020), **SynSetExpan** (Shen et al., 2020a), **ProbExpan** (Li et al., 2022), and **Llama-3.1 70B** (Dubey et al., 2024). Additionally, since TAXOINSTRUCT is pre-trained on both parent-finding and sibling-finding tasks, we investigate whether the former enhances the latter. To assess this, we introduce an ablation variant, **NoParentPretrain**, which is pre-trained on the sibling-finding task only.

Evaluation Metric. Following previous studies (Shen et al., 2017; Yan et al., 2019; Zhang et al., 2020), we adopt the Mean Average Precision (**MAP@**k) as the evaluation metric.

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 ▷: 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 ▷	0.985*	0.990*	0.991*	0.978*
ProbExpan ▷	0.993	0.990*	0.995	0.982
Llama-3.1 70B	0.9933	0.9788*	0.9861*	0.9748*
TAXOINSTRUCT	**0.9956**	**0.9928**	**0.9957**	**0.9875**
NoParentPretrain	0.9867*	0.9689*	0.9746*	0.9720*

Implementation Details. We initialize our model with Llama-3 8B (Dubey et al., 2024) and continuously pre-train/fine-tune it using Low-Rank Adaptation (LoRA) (Hu et al., 2022). The optimizer is AdamW (Loshchilov and Hutter, 2017), and the batch size is 64. We adopt SPECTER (Cohan et al., 2020) as the pre-trained encoder $\mathrm{E}(\cdot)$ in Eqs. (1) and (2).

Experimental Results. Table 1 presents the MAP@10 and 20 scores of compared methods in the entity set expansion task. We run TAXOINSTRUCT multiple times and report the average performance. To assess statistical significance, we conduct a two-tailed Z-test comparing TAXOINSTRUCT against each baseline, with significance levels indicated in Table 1. We can observe that: (1) TAXOINSTRUCT consistently outperforms all baselines, including those leveraging 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, suggesting that even in entity set expansion—where identifying siblings is the primarily required skill—pre-training TAXOINSTRUCT to find parents still enhances the performance. This finding validates our motivation for pre-training a unified model to jointly address different yet related tasks.

4.2 Taxonomy Expansion

Datasets. Following (Jiang et al., 2023b), we use two benchmark datasets, **Environment** and **Science**, from the shared task in SemEval 2016 (Bordea et al., 2016). Entities in these two datasets are scientific concepts related to environment and general science, respectively.

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*
Llama-3.1 70B	0.3654*	0.6957*	0.4471*	0.7310*
TAXOINSTRUCT	**0.5115**	**0.8300**	**0.6165**	0.8480
NoSiblingPretrain	0.4616*	0.7911*	0.5953*	**0.8559**

Baselines. The baselines for taxonomy expansion include **TAXI** (Panchenko et al., 2016), **HypeNET** (Shwartz et al., 2016), **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), and **Llama-3.1 70B** (Dubey et al., 2024). Additionally, to investigate if sibling finding helps parent finding, we introduce an ablation version of TAXOINSTRUCT, **NoSiblingPretrain**, for the taxonomy expansion task, which is pre-trained on the parent-finding task only.

Evaluation Metrics. We adopt Accuracy (**Acc**) and Wu & Palmer Similarity (**Wu&P**) (Wu and Palmer, 1994) as the evaluation metrics. 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. Our key observations are: (1) TAXOINSTRUCT significantly outperforms all baselines in nearly every case. The only exception is that TEMP achieves a higher Wu&P score on the Science dataset. Apart from TEMP, GenTaxo is a strong baseline that follows a generative paradigm for taxonomy expansion. However, unlike TAXOINSTRUCT, which leverages LLMs to fully harness

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	Parent nDCG	Sibling nDCG	Parent nDCG
HSetExpan	0.8814*	0.8268*	0.6515*	0.5085*
NoREPEL	0.8830*	0.8152*	0.6705*	0.6216*
NoGTO	0.9527*	0.8855*	0.7395*	0.6428*
HiExpan	0.9524*	0.9045	0.7365*	0.7132*
Llama-3.1 70B	0.9708*	0.8607*	0.8934*	0.8010
TaxoInstruct	**0.9817**	**0.9210**	**0.9220**	**0.8034**
NoParentPretrain	0.9668*	0.7836*	0.8920*	0.7864
NoSiblingPretrain	0.9425*	0.9114	0.7930*	0.6838*

the strengths of the generative approach, GenTaxo relies solely on a Gated Recurrent Unit (GRU) architecture, resulting in suboptimal performance. (2) TaxoInstruct outperforms NoSiblingPretrain across most columns, suggesting that even in the taxonomy expansion task—where identifying parent entities is the primary objective—pre-training the model to accurately identify siblings remains beneficial. Combined with our ablation analysis from entity set expansion, this finding supports the conclusion that sibling-finding and parent-finding skills can *mutually* enhance each other.

4.3 Seed-Guided Taxonomy Construction

Datasets. We use the **DBLP** and **PubMed-CVD** datasets introduced by Shen et al. (2018a). The seeds in our experiments are identical to those in Shen et al. (2018a). Both datasets have a two-layer input taxonomy. 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. 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.

Baselines. The baselines for seed-guided taxonomy construction include **HSetExpan** (Shen et al., 2017), **HiExpan** (Shen et al., 2018a), two ablation versions of HiExpan—**NoREPEL** (Shen et al., 2018a) and **NoGTO** (Shen et al., 2018a)—as well as **Llama-3.1 70B** (Dubey et al., 2024). Besides, following our practice in the previous two tasks, we consider two ablation variants, **NoParentPretrain** and **NoSiblingPretrain**.

Evaluation Metrics. At the top layer, both our TaxoInstruct model and most baselines achieve near-perfect accuracy. Therefore, our evaluation focuses on the more challenging bottom layer. We use **Sibling nDCG@k** to assess the accuracy of the sibling-finding step and **Parent nDCG@k** to evaluate the accuracy of the parent-finding step.

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	0.9708	0.8934	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**

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 clearly outperforms all baselines in both the sibling-finding and parent-finding steps across both datasets. Notably, identifying correct sibling terms that are relevant to the taxonomy is a prerequisite for accurately determining their parent categories. If an expanded sibling is incorrect (i.e., it does not belong at this layer or anywhere in the taxonomy), predicting its correct parent becomes impossible. This explains why the Sibling nDCG@50 score is always higher than the corresponding Parent nDCG@50 score. (2) TaxoInstruct consistently outperforms the two ablation versions, which is intuitive, as seed-guided taxonomy construction relies on the synergy of both skills.

4.4 Effect of the LLM Backbone

Although we use Llama-3 8B as the backbone for TaxoInstruct in previous experiments, it is important to emphasize that TaxoInstruct is a versatile framework that can be instantiated with various off-the-shelf generative LLMs. To demonstrate the generalizability of TaxoInstruct, we evaluate its performance when Llama-2-chat 7B (Touvron et al., 2023), Mistral 7B (Jiang et al., 2023a), and Gemma 7B (Team et al., 2024) are plugged in.

Table 4 presents the performance of TaxoInstruct with different LLM backbones. Due to space limitations, we only display results for 4 datasets (out of the 6 used in the previous experiments) and one metric for each dataset. From Table 4, we observe that: (1) On DBLP, both Llama-3 8B and Llama-2-chat 7B allow us to outperform the strongest baseline—Llama-3.1 70B, which has a much larger number of parameters; on PubMed-

CVD, this could be achieved using Llama-3 8B and Mistral 7B. (2) On the Environment dataset, both Llama-3 8B and Gemma 7B enable our framework to beat the best-performing baseline (i.e., TEMP). On the Science dataset, even our default choice Llama-3 8B does not perform the best in Table 3, using Gemma 7B allows us to surpass 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.

5 Related Work

Entity Set Expansion. EgoSet (Rong et al., 2016) pioneers entity set expansion using skip-grams and word2vec embeddings (Mikolov et al., 2013). Following this, SetExpan (Shen et al., 2017) employs an iterative bootstrapping framework, while CaSE (Yu et al., 2019) rank candidates via distributional similarity among context-free embeddings to rank candidate entities according to the seeds. With pre-trained contextualized language models such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), CGExpan (Zhang et al., 2020) generates class names to prevent semantic drift, ProbExpan (Li et al., 2022) refines entity representations using contrastive learning, and GAPA (Li et al., 2023) leverages autoregressive models for context pattern generation. 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 et al., 2016) are used to infer the hypernym-hyponym relationship. Later, 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. In addition, TMN (Zhang et al., 2021) examines candidate parents and children via a triplet matching network. Most recently, TaxoPrompt (Xu et al., 2022) and TacoPrompt (Xu et al., 2023) adopt prompt tuning on BERT-based encoder models 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-3. More recently, there are studies (Zeng et al., 2024a,b) leveraging GPT-4 and advanced prompting techniques for taxonomy expansion. By contrast, 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 Prompting and Instruction Tuning. There has been increasing attention on prompting and instruction-tuning LLMs to learn from (text-rich) structured data (Jin et al., 2023; Li et al., 2024; Chen et al., 2024). For instance, Wang et al. (2023a) strategically prompt LLMs to solve graph problems such as shortest paths and maximum flows; InstructGLM (Ye et al., 2024) shows that LLMs fine-tuned on node classification and link prediction can outperform competitive graph neural network baselines; Zhang et al. (2024) put entity triplets into an instruction template for LLMs to perform knowledge graph completion; Guo et al. (2023) conduct a benchmark study on LLMs' ability to understand graph data by using formal language to describe graphs. Different from these studies that focus on graph structures (e.g., academic 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.

6 Conclusions

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

Limitations

Our work has the following limitations. First, since our primary goal is to verify the universal effectiveness of LLM instruction tuning across all three tasks, we intentionally keep our framework as simple as possible, avoiding complex signals utilized in previous studies such as paths (Liu et al., 2021; Jiang et al., 2022) and local graphs (Mao et al., 2020; Wang et al., 2021). Second, after instruction tuning, TAXOINSTRUCT can be further equipped with inference-time techniques such as chain-of-thought prompting (Wei et al., 2022b) and self-consistency reasoning (Wang et al., 2023b). Integrating these techniques into TAXOINSTRUCT could further enhance its performance, which we leave for future work.

References

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *SemEval'16*, pages 1081–1091.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In *ACL'20*, pages 2270–2282.

Allan Peter Davis, Thomas C Wiegers, Robin J Johnson, Daniela Sciaky, Jolene Wiegers, and Carolyn J Mattingly. 2022. Comparative toxicogenomics database (ctd): update 2023. *Nucleic Acids Research*, 51(D1):D1257–D1262.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT'19*, pages 4171–4186.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Rafael S Gonçalves, Matthew Horridge, Rui Li, Yu Liu, Mark A Musen, Csongor I Nyulas, Evelyn Obamos, Dhananjay Shrouty, and David Temple. 2019. Use of owl and semantic web technologies at pinterest. In *ISWC'19*, pages 418–435.

Jiayan Guo, Lun Du, and Hengyu Liu. 2023. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR'18*.

Jiaxin Huang, Yiqing Xie, Yu Meng, Jiaming Shen, Yunyi Zhang, and Jiawei Han. 2020. Guiding corpus-based set expansion by auxiliary sets generation and co-expansion. In *WWW'20*, pages 2188–2198.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. 2022. Taxoenrich: Self-supervised taxonomy completion via structure-semantic representations. In *WWW'22*, pages 925–934.

Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. 2023b. A single vector is not enough: Taxonomy expansion via box embeddings. In *WWW'23*, pages 2467–2476.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.

Yinghui Li, Shulin Huang, Xinwei Zhang, Qingyu Zhou, Yangning Li, Ruiyang Liu, Yunbo Cao, Hai-Tao Zheng, and Ying Shen. 2023. Automatic context pattern generation for entity set expansion. *IEEE TKDE*, 35(12):12458–12469.

Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive learning with hard negative entities for entity set expansion. In *SIGIR'22*, pages 1077–1086.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2024. A survey of graph meets large language model: Progress and future directions. In *IJCAI'24*, pages 8123–8131.

Zichen Liu, Hongyuan Xu, Yanlong Wen, Ning Jiang, Haiying Wu, and Xiaojie Yuan. 2021. Temp: taxonomy expansion with dynamic margin loss through taxonomy-paths. In *EMNLP'21*, pages 3854–3863.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *ICLR'19*.

Jonathan Mamou, Oren Pereg, Moshe Wasserblat, Ido Dagan, Yoav Goldberg, Alon Eirew, Yael Green, Shira Guskin, Peter Izsak, and Daniel Korat. 2018. Setexpander: End-to-end term set expansion based on multi-context term embeddings. In *COLING'18 System Demonstrations*, pages 58–62.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *WWW'20*, pages 2044–2054.

Yuning Mao, Tong Zhao, Andrey Kan, Chenwei Zhang, Xin Luna Dong, Christos Faloutsos, and Jiawei Han. 2020. Octet: Online catalog taxonomy enrichment with self-supervision. In *KDD'20*, pages 2247–2257.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS'13*, pages 3111–3119.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS'22*, pages 27730–27744.

Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *SemEval'16*, pages 1320–1327.

Meng Qu, Xiang Ren, Yu Zhang, and Jiawei Han. 2018. Weakly-supervised relation extraction by pattern-enhanced embedding learning. In *WWW'18*, pages 1257–1266.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Xin Rong, Zhe Chen, Qiaozhu Mei, and Eytan Adar. 2016. Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In *WSDM'16*, pages 645–654.

Chao Shang, Sarthak Dash, Md Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, and Alfio Gliozzo. 2020. Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer. In *ACL'20*, pages 2198–2208.

Jiaming Shen, Wenda Qiu, Jingbo Shang, Michelle Vanni, Xiang Ren, and Jiawei Han. 2020a. Synsetexpan: An iterative framework for joint entity set expansion and synonym discovery. In *EMNLP'20*, pages 8292–8307.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020b. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *WWW'20*, pages 486–497.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *ECML-PKDD'17*, pages 288–304.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T Vanni, Brian M Sadler, and Jiawei Han. 2018a. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *KDD'18*, pages 2180–2189.

Zhihong Shen, Hao Ma, and Kuansan Wang. 2018b. A web-scale system for scientific knowledge exploration. In *ACL'18 System Demonstrations*, pages 87–92.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *ACL'16*, pages 2389–2398.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *ACL'18*, pages 263–272.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023a. Can language models solve graph problems in natural language? In *NeurIPS'23*.

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

Suyuchen Wang, Ruihui Zhao, Xi Chen, Yefeng Zheng, and Bang Liu. 2021. Enquire one's parent and child before decision: Fully exploit hierarchical structure for self-supervised taxonomy expansion. In *WWW'21*, pages 3291–3304.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *ICLR'23*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *ICLR'22*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS'22*, pages 24824–24837.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *ACL'94*, pages 133–138.

Hongyuan Xu, Yunong Chen, Zichen Liu, Yanlong Wen, and Xiaojie Yuan. 2022. Taxoprompt: A prompt-based generation method with taxonomic context for self-supervised taxonomy expansion. In *IJCAI'22*, pages 4432–4438.

Hongyuan Xu, Ciyi Liu, Yuhang Niu, Yunong Chen, Xiangrui Cai, Yanlong Wen, and Xiaojie Yuan. 2023. Tacoprompt: A collaborative multi-task prompt learning method for self-supervised taxonomy completion. In *EMNLP'23*, pages 15804–15817.

Lingyong Yan, Xianpei Han, Le Sun, and Ben He. 2019. Learning to bootstrap for entity set expansion. In *EMNLP'19*, pages 292–301.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2024. Language is all a graph needs. In *Findings of EACL 2024*, pages 1955–1973.

Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan. 2019. Corpus-based set expansion with lexical features and distributed representations. In *SIGIR'19*, pages 1153–1156.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *KDD'20*, pages 1026–1035.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. 2024a. Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. In *CIKM'24*, pages 3093–3102.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Zhenyu Wu, Shangbin Feng, and Meng Jiang. 2024b. Codetaxo: Enhancing taxonomy expansion with limited examples via code language prompts. *arXiv preprint arXiv:2408.09070*.

Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *KDD'21*, pages 2104–2113.

Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. 2021. Taxonomy completion via triplet matching network. In *AAAI'21*, pages 4662–4670.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. 2024. Making large language models perform better in knowledge graph completion. In *ACM MM'24*, pages 233–242.

Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via language model probing. In *ACL'20*, pages 8151–8160.

A Appendix

A.1 Details of Baselines

For all three tasks, we use Llama-3.1 70B (Dubey et al., 2024) as one of our baselines, which is directly prompted with the same instructions as TAXOINSTRUCT. Besides, we consider the following task-specific baselines.

A.1.1 Baselines of Entity Set Expansion

- **EgoSet** (Rong et al., 2016) uses skip-gram context features and word2vec embeddings to expand entity sets in multiple facets.
- **SetExpan** (Shen et al., 2017) iteratively selects skip-gram context features from the corpus and proposes a rank ensemble mechanism for scoring and selecting entities.
- **SetExpander** (Mamou et al., 2018) learns different text embeddings from different types of context features and trains a classifier to predict whether an entity belongs to a set.
- **CaSE** (Yu et al., 2019) integrates skip-grams and word2vec embeddings to score and rank entities from the corpus.
- **SetCoExpan** (Huang et al., 2020) generates auxiliary sets as negative sets that are closely related to the target set and simultaneously co-expand multiple sets.
- **CGExpan** (Zhang et al., 2020) infers the target semantic class names by probing a language model and then utilizes the generated class names to expand new entities.
- **SynSetExpan** (Shen et al., 2020a) jointly conducts two related tasks—synonym discovery and entity set expansion—and utilizes synonym information to enhance expansion performance.
- **ProbExpan** (Li et al., 2022) devises an entity-level masked language model with contrastive learning to refine the representation of entities for entity set expansion.

A.1.2 Baselines of Taxonomy Expansion

- **TAXI** (Panchenko et al., 2016) first extracts hypernym-hyponym pairs from text corpora using substrings and lexico-syntactic patterns, then it organizes the extracted terms into a coherent taxonomy.

- **HypeNET** (Shwartz et al., 2016) employs LSTM to concurrently capture the distributional and relational information between term pairs along dependency paths.

- **BERT+MLP** (Devlin et al., 2019) first acquires term embeddings from a pre-trained BERT model and then inputs the embeddings into a multi-layer perceptron to predict the hypernymy relationship.

- **TaxoExpan** (Shen et al., 2020b) leverages graph neural networks to encode local ego-graphs in the input taxonomy to improve entity representations. In the original paper, context-free word embeddings are used as input features. Following (Yu et al., 2020), we replace context-free embeddings with more powerful BERT embeddings for this baseline.

- **Arborist** (Manzoor et al., 2020) explores heterogeneous edge semantics by employing a large-margin ranking loss to ensure an upper limit on the shortest-path distance between predicted and actual parent nodes.

- **Graph2Taxo** (Shang et al., 2020) utilizes cross-domain graph structures and constraint-based learning of directed acyclic graphs.

- **STEAM** (Yu et al., 2020) learns representations for each pair of (new entity, existing entity) from multiple views using paths sampled from the taxonomy.

- **TMN** (Zhang et al., 2021) proposes a triplet matching network to match a query with hypernym-hyponym pairs. It enables insertion of non-leaf query concepts into an existing taxonomy.

- **TEMP** (Liu et al., 2021) employs pre-trained contextual encoders to predict the position of new concepts by ranking the generated taxonomy-paths.

- **GenTaxo** (Zeng et al., 2021) learns the contextual embeddings from their surrounding graph-based and language-based relational information and leverages the corpus for pre-training a concept name generator.

- **BoxTaxo** (Jiang et al., 2023b) represents entities as boxes to capture their parent-child relationship. It optimizes the box embedding (Vilnis et al., 2018) of each entity from a joint view of geometry and probability.

A.1.3 Baselines of Seed-Guided Taxonomy Construction

- **HSetExpan** (Shen et al., 2017) iteratively applies SetExpan at each layer of the input taxonomy. For each expanded bottom-layer node, it uses REPEL (Qu et al., 2018), a weakly supervised relation extraction model, to find the most proper parent at the top layer.

- **HiExpan** (Shen et al., 2018a) combines the techniques of flat set expansion, parent-child relationship inference, and global optimization of the taxonomy structure by jointly utilizing skip-grams, context-free text embeddings, and entity types.

- **HiExpan-NoREPEL** (Shen et al., 2018a) is an ablation version of HiExpan, which does not utilize REPEL for parent-child relationship inference. Instead, it uses context-free text embeddings only.

- **HiExpan-NoGTO** (Shen et al., 2018a) is an ablation version of HiExpan, which does not have the global optimization optimization module.

Shen et al. (2018a) have released the output taxonomies[2] of the four baselines above on DBLP and PubMed-CVD, which we use for evaluation.

A.2 Details of Evaluation Metrics

A.2.1 Metric for Entity Set Expansion

We use **MAP@k** as the evaluation metric. 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

$$\text{AP@}k(\mathcal{S}, \mathcal{S}^+) = \frac{1}{k} \sum_{\substack{i:1 \le i \le k, \\ s_{M+i} \sim \mathcal{S}}} \frac{\sum_{j=1}^{i} \mathbb{I}(s_{M+j} \sim \mathcal{S})}{i}. \quad (3)$$

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 $\mathcal{S}_1^+, ..., \mathcal{S}_C^+$, the MAP@$k$ score is defined as

$$\text{MAP@}k = \frac{1}{C} \sum_{i=1}^{C} \text{AP@}k(\mathcal{S}_i, \mathcal{S}_i^+). \quad (4)$$

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

A.2.2 Metrics for Taxonomy Expansion

We use Accuracy (**Acc**) and Wu & Palmer Similarity (**Wu&P**) as the evaluation metrics.

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 $\textsc{Parent}^+(\cdot)$ is defined as

$$\text{Acc} = \frac{1}{C}\sum_{i=1}^{C} \mathbb{I}(\textsc{Parent}^+(x_i) = y_i). \tag{5}$$

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.

$$\text{Wu\&P} = \frac{1}{C}\sum_{i=1}^{C} \frac{2 \times \text{depth}(\text{LCP}(\textsc{Parent}^+(x_i), y_i))}{\text{depth}(\textsc{Parent}^+(x_i)) + \text{depth}(y_i)}, \tag{6}$$

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.

A.2.3 Metrics for Seed-Guided Taxonomy Construction

We use **Sibling nDCG@k** and **Parent nDCG@k** as the evaluation metrics. Formally, in a two-layer taxonomy, given the bottom-layer seeds $\mathcal{S}_2 = \{s_{2,1}, ..., s_{2,M}\}$, we examine the top-$k$ expanded bottom-layer entities $\mathcal{S}_2^+ = \{s_{2,M+1}, ..., s_{2,M+k}\}$. 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).

$$\text{Sibling nDCG@}k = \frac{\sum_{i=1}^{k} \frac{\mathbb{I}(s_{2,M+i} \sim \mathcal{S}_2)}{\log_2(i+1)}}{\sum_{i=1}^{k} \frac{1}{\log_2(i+1)}}. \tag{7}$$

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

$$\text{Parent nDCG@}k = \frac{\sum_{i=1}^{k} \frac{\mathbb{I}(\textsc{Parent}^+(s_{2,M+i}) = s_{1,p(i)})}{\log_2(i+1)}}{\sum_{i=1}^{k} \frac{1}{\log_2(i+1)}}. \tag{8}$$