Chain-of-Layer: Iteratively Prompting Large Language Models for Taxonomy Induction from Limited Examples

Qingkai Zeng*†
qzeng@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Yuyang Bai*
ybai3@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Zhaoxuan Tan
ztan3@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Shangbin Feng
shangbin@cs.washington.edu
University of Washington
Seattle, WA, USA

Zhenwen Liang
Zhihan Zhang
zliang6@nd.edu
zzhang23@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Meng Jiang
mjiang2@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Abstract

Automatic taxonomy induction is crucial for web search, recommendation systems, and question answering. Manual curation of taxonomies is expensive in terms of human effort, making automatic taxonomy construction highly desirable. In this work, we introduce Chain-of-Layer which is an in-context learning framework designed to induct taxonomies from a given set of entities. Chain-of-Layer breaks down the task into selecting relevant candidate entities in each layer and gradually building the taxonomy from top to bottom. To minimize errors, we introduce the Ensemble-based Ranking Filter to reduce the hallucinated content generated at each iteration. Through extensive experiments, we demonstrate that Chain-of-Layer achieves state-of-the-art performance on four real-world benchmarks. Source code available at: https://github.com/qingkaizeng/chain-of-layer.

CCS Concepts

• **Computing methodologies** → **Information extraction**.

Keywords

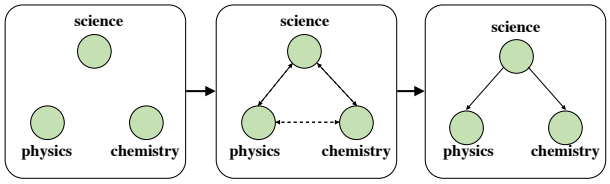
Taxonomy Induction; Large Language Models; In-context Learning

ACM Reference Format:
Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. 2024. Chain-of-Layer: Iteratively Prompting Large Language Models for Taxonomy Induction from Limited Examples. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679608

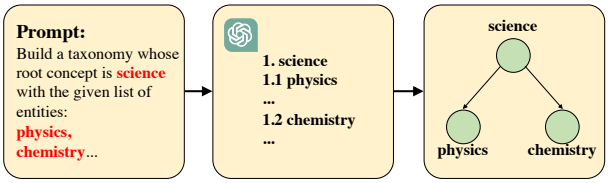
*Equal contribution.
†Corresponding Author

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679608



(a) Discriminative Methods: Scoring each entity pair and pruning to taxonomic structure [6, 26]



(b) Generative Methods: Prompting LLMs to generate taxonomy

Figure 1: Two Types of Methods for Taxonomy Induction

1 Introduction

Taxonomy refers to a hierarchical structure that outlines the connections between concepts or entities. It commonly represents these relationships through hypernym-hyponym associations or "is-a" relationships. Taxonomies are essential in aiding several tasks, such as textual content understanding [10, 16, 34], personalized recommendations [12, 31, 39], and questions answering [36]. However, developing a taxonomy solely based on human experts can be a time-consuming and costly process, often presenting challenges in terms of scalability. Consequently, recent efforts have focused on automatic taxonomy induction, which aims to autonomously organize a group of entities into a taxonomy.

Traditional approaches in taxonomy induction follow the discriminative method illustrated in Figure 1a and aim to identify and structure parent-child relations among entities in a hierarchical manner. Early efforts involve learning these relations by leveraging the semantic connections between entities. The semantics can be represented by lexical patterns [17, 24, 29, 38], distributional word embeddings [9, 20, 26, 28], and contextual pre-trained models [6, 13]. Following this, the identified relations are organized into a taxonomic structure using various pruning techniques [2, 21, 24, 32].

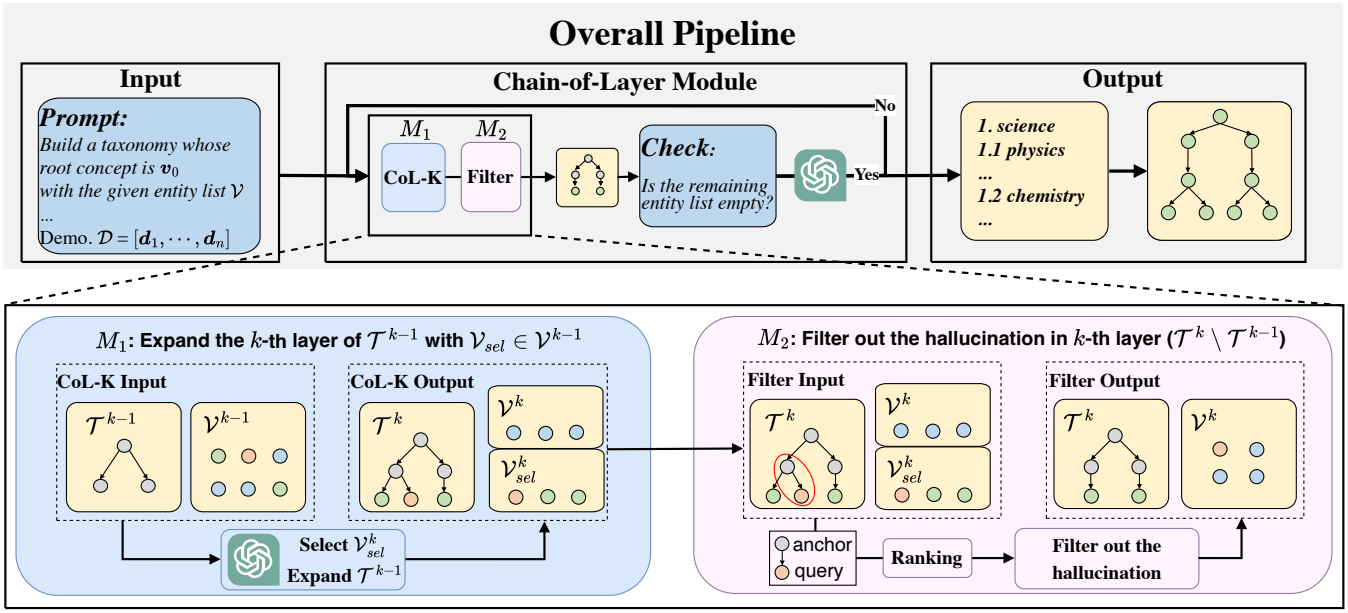


Figure 2: The overview of the framework for Chain-of-Layer (CoL): Given an entity list $\mathcal{V}$ and a root entity $v_0 \in \mathcal{V}$, CoL systematically organizes the entities in $\mathcal{V}$ into hierarchical groups, incrementally adding them to the taxonomy in a top-down manner at each iteration. In detail, at the k-th iteration, CoL-K selects a subset of entities $\mathcal{V}$sel from the k-level and extends the existing taxonomy $\mathcal{T}^{k-1}$ with these entities. The newly generated parent-child relations ($\mathcal{T}^k \setminus \mathcal{T}^{k-1}$) are refined by an Ensemble-based Ranking Filter to reduce the hallucinations into the output taxonomy $\mathcal{T}^k$ in k-th iteration. The process continues until all entities in $\mathcal{V}$ are integrated into the resulting taxonomy.

Recently, Large Language Models (LLMs) have shown impressive skills in understanding and generating text, enabling them to adapt to a wide range of domains and tasks [1, 23]. Consequently, many studies have been conducted to leverage the capabilities of LLMs for Information Extraction (IE) tasks using a generative approach [35]. Furthermore, increasing the number of parameters of LLMs significantly enhances their ability to generalize, surpassing smaller pre-trained models, and enabling them to deliver outstanding performance in few-shot or zero-shot settings [15]. Figure 1b illustrates the pipeline depicting how generative methods operate on the taxonomy induction task.

In the context of taxonomy induction with large language models, TaxonomyGPT [5] first attempts to prompt LLMs to predict the hierarchical relation among the given concepts. However, TaxonomyGPT shows two major limitations in taxonomy induction. First, it ignores the inherent structure of taxonomies during the generation of new parent-child relations. The reason is that TaxonomyGPT produces parent-child relations among given entities independently, leading to the loss of crucial taxonomic structure information, such as sibling-sibling and ancestor-descendant relations. Consequently, this neglect results in structural inaccuracies in the output taxonomy, including the emergence of multiple root entities and circular relations. Second, as with all the methods based on prompting LLMs, TaxonomyGPT also suffers from the issue of hallucination. For example, even though we have highlighted the requirement in the instruction, LLMs still add entities that are not related to the target taxonomy into the output taxonomy.

To address the above issues, we first introduce HF, **H**ierarchical **F**ormat Taxonomy Induction Instruction to represent taxonomic structures via hierarchical numbering format. For example, as shown in Figure 1b, science is the sole root entity, so it is indexed as '1. science'. Physics and chemistry, being child entities of science, are thus indexed as '1.1 physics' and '1.2 chemistry'. This format ensures that each entity within the taxonomy possesses a global view of its hierarchical structure, like physics is the sibling entity of chemistry since they share the same hierarchical format ('1.x'). It is important to note that all the methods proposed in this work adhere to HF for representing the taxonomic structure.

Second, to reduce the hallucination generated by the inductive process, we propose the Chain-of-Layer (CoL) unlike prompting LLMs to generate target taxonomy in one iteration, CoL decomposes the taxonomy induction task in a layer-to-layer manner. Specifically, for each iteration, CoL selects a subset of entities from the given entity set and expands the current taxonomy with these selected entities. The key insight of this decomposition is to instruct the LLMs to explicitly anchor each of their reasoning iterations in the taxonomy induction task. Benefits on the iterative setting of CoL, we incorporate an Ensemble-based Ranking Filter at each iteration as a post-processing module to reduce the error propagation from the current iteration to the next iteration. We also develop CoL-Zero to extend CoL to zero-shot settings where annotated taxonomies are unavailable. CoL-Zero uses LLMs to generate taxonomies as demonstrations instead of relying on human annotations.

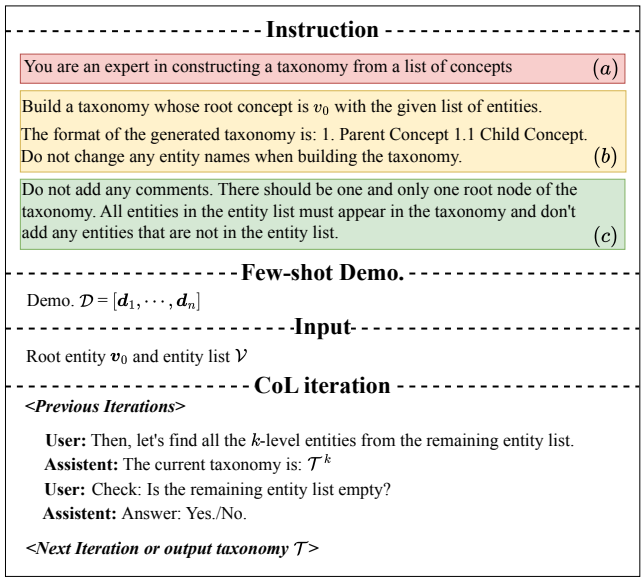


Figure 3: Prompt Overview of Chain-of-Layer Framework

The efficacy of HF and CoL has been validated through extensive experiments on WordNet sub-taxonomies and three large-scale, real-world taxonomies. The results demonstrate that both HF and CoL outperform all baseline methods across multiple evaluation metrics. We also explore the performance of CoL-Zero on the benchmarks mentioned above. Some interesting observations of CoL-Zero are presented in this work.

In summary, this study makes the following contributions:

- We introduce HF, the Hierarchical Format Taxonomy Induction Instruction, to utilize the hierarchical structure of the entities to increase the quality of the inducted taxonomy.
- We introduce Chain-of-Layer (CoL), an iterative taxonomy induction framework that incorporates the Ensemble-based Ranking Filter for reducing the hallucinations in the output taxonomies generated by LLMs.
- Extensive experiments demonstrate that HF and CoL significantly improve the performance of taxonomy induction tasks on four datasets from various domains.

Scope and Limitation. This study represents an initial effort to utilize LLMs for taxonomy induction. Our main focus is to identify an effective in-context learning framework to harness the capabilities of LLMs. We are aware that the performance of our proposed approach on large-scale taxonomies is constrained by the limitations in instruction-following capabilities and the context window size of LLMs. However, how to facilitate the ability of LLMs to handle extremely long prompts is beyond the scope of this paper. We hope this work will inspire future research in this area.

2 Problem Definition

We define a taxonomy, denoted as $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, as a directed acyclic graph composed of two components: a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$. In the task of taxonomy induction, the model is provided with a set of conceptual entities, represented by $\mathcal{V}$, where each entity can be either a single word or a short phrase. The objective is to construct the taxonomy $\mathcal{T}$ based on these given entities.

3 Methodology

In this section, we provide a comprehensive overview of our proposed Chain-of-Layer (CoL) framework designed for addressing the taxonomy induction task. Specifically, CoL dissects the taxonomy induction task through a layer-to-layer approach. As shown in Figure 3, Our CoL framework consists of four parts: instruction (HF, **H**ierarchical **F**ormat Taxonomy Induction Instruction), few-shot demonstration, input, and CoL iteration. In the instruction part (Sec. 3.1), we configure the system message of the LLM, specify the objectives of the task and the expected output format, and establish a series of rules that help the model understand and accurately complete the task. In Sec. 3.2, we describe and formalize the process of inducting our demonstrations. In Sec. 3.3, we introduce the iterative process of CoL and the Ensemble-based Ranking Filters tailored to mitigate hallucinations that may arise during the process. Finally, in Sec. 3.4, we extend our CoL to the zero-shot setting. The details of each module in CoL are presented in Figure 2.

3.1 Hierarchical Format Taxonomy Induction Instruction (HF)

To enable LLMs to more effectively and accurately complete the taxonomy induction task, we propose HF, the **H**ierarchical **F**ormat Taxonomy Induction Instruction. As shown in Figure 3, the instruction specifies the objectives of the taxonomy induction task, which can be decomposed into three components. In component (a), LLMs are instructed to utilize the domain expertise to generate the desired output. Component (b) provides instructions for the output format, which is expected to adhere to a hierarchical numbering format. This format ensures that each entity within the generated taxonomy possesses a comprehensive understanding of its hierarchical structure. Finally, component (c) highlights a set of fundamental rules $\mathcal{R}$ about the taxonomy induction task. These rules include: 1. Do not use entities not covered in the given entity set and ensure that all entities listed in the given entity list are present in the taxonomy ($\boldsymbol{r}_1$); 2. Maintain a single root entity within the taxonomy ($\boldsymbol{r}_2$); 3. Refrain from adding comments ($\boldsymbol{r}_3$).

3.2 Few-shot Demonstration Construction

To enable the model to better follow our instructions to complete the task, we propose a method for constructing demonstrations for CoL inference. For each demonstration d_i, we decompose each taxonomy $\mathcal{T}_i$ in hierarchical order and simulate the process of inducting the entire taxonomy from top to bottom. At the end of each level of induction, we prompt LLM whether the current taxonomy has included all entities from the given entity set. If a negative response is received, we will continue to expand the current taxonomy layer downward until it encompasses all entities from the given entity set. The demonstration d_i employed for expanding the k-th layer of the demo taxonomy $\mathcal{T}_{\boldsymbol{d}_i}^{k-1}$ are presented as follows:

<messages of previous iteration>
Assistant: *The current taxonomy is:* $\mathcal{T}_{\boldsymbol{d}_i}^{k-1}$
User: *Check: Is the remaining entity list empty?*
Assistant: *Answer: No.*
User: *Then, let's find all the k-th level entities from the remaining entity list.*

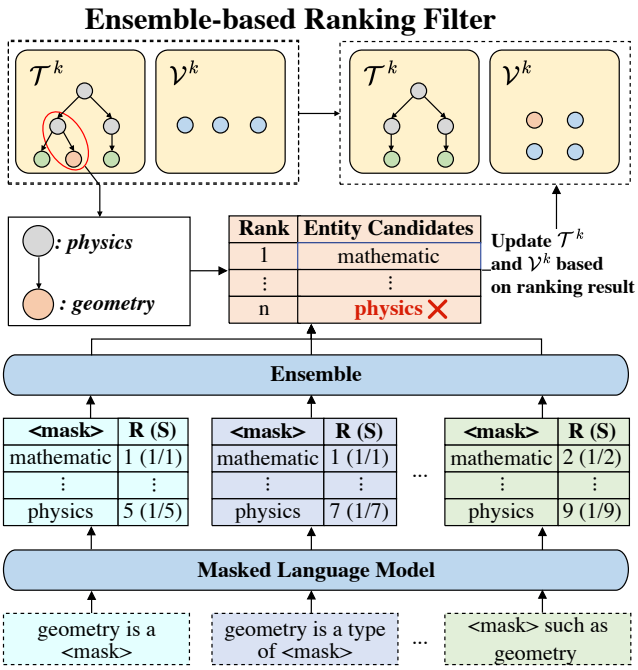


Figure 4: The details of the Ensemble-based Ranking Filter.

Assistant: *The current taxonomy is:* $\mathcal{T}^k_{\boldsymbol{d}_i} \leftarrow \mathcal{T}^{k-1}_{\boldsymbol{d}_i} \bigcup \mathcal{V}^k_{sel_i}$
<beginning of next iteration>

We use the first five sub-taxonomies from WordNet's training set as demonstrations $\mathcal{D}^{(t)} = [\boldsymbol{d}_1^{(t)}, \cdots, \boldsymbol{d}_5^{(t)}]$ in this case for fair comparison.

3.3 Inference via Chain-of-Layer

3.3.1 Ensemble-based Ranking Filter. It is well-known that large language models are greatly affected by hallucinations during the process of generating target texts, resulting in content that is significantly different from the target [14]. In the taxonomy induction task, we mainly observe two categories of LLM hallucinations: (1) The large language models do not strictly use the entities in the given entity set but instead include non-target entities in the output taxonomy; (2) The large language models introduce incorrect parent-child relations into the output taxonomy.

To alleviate these issues, we propose a filter module in the CoL framework. Specifically, in the process of inducting each layer of the taxonomy, the filter removes incorrect parent-child relations in each iteration of the model's output, preventing the error caused by hallucination propagating to the next iteration.

Our filter design is based on an ensemble mechanism. We propose a set of templates $\mathcal{M}$ and used a pre-trained mask language model to rank the generated parent-child relations in each iteration for all templates $\boldsymbol{m} \in \mathcal{M}$. We present the $\mathcal{M}$ as follows:

<query> *is a/an* <anchor>
<query> *is a kind of* <anchor>
<query> *is a type of* <anchor>
<query> *is an example of* <anchor>
<anchor> *such as* <query>
A/An <anchor> *such as* <query>

For each entity $\boldsymbol{q}$ in the entity list, we compute the probability of tokens at the <anchor> position by placing $\boldsymbol{q}$ in the <query> position (as a child entity) of template $\boldsymbol{m}$ and positioning one of the remaining entities $\boldsymbol{a}$ at the <anchor> (as a parent entity). Then we sort the token probabilities to determine the similarity ranking. Subsequently, inference, represented by $\text{Sim}(\boldsymbol{q}, \boldsymbol{a}|\boldsymbol{m})$, is computed utilizing the reciprocal of this similarity ranking. Finally, we ensemble the scoring results of each template to obtain the final filter score. The formula of the similarity score is:

$$score(\boldsymbol{q}|\boldsymbol{a}, \mathcal{M}, \mathcal{V}) = \frac{1}{|\mathcal{M}|} \sum_{\boldsymbol{m} \in \mathcal{M}} \text{Sim}(\boldsymbol{q}, \boldsymbol{a}|\boldsymbol{m}) \tag{1}$$

For each query entity, we retain only the top ten parent candidates. If the parent-child relations output by the LLM are not within this range, these relations will be filtered out. In this paper, we use a pre-trained masked language model specialized for the scientific domain and tasks called SciBERT [3] to ensure that the pre-trained models contain sufficient domain knowledge to complete the ranking process.

3.3.2 Iterative Inference. After providing the instructions and constructing the few-shot demonstrations, we introduce the interactive inference process of our CoL framework with the Ensemble-based Ranking Filter. We provide the input entity candidates set $\mathcal{V}$ and the initial taxonomy $\mathcal{T}^0$ only including root entity v_0 to the LLM and expect the LLM to generate the output taxonomy $\mathcal{T}$ according to the rules $\mathcal{R}$ and the defined format of the demonstrations $\mathcal{D}$ in section 3.2. The whole inference process in CoL is donated as:

$$\mathcal{T} = \text{CoL}(\mathcal{V}, \mathcal{T}^0, \mathcal{D}, \mathcal{R}) \tag{2}$$

The inference starts with $k = 0$ and define $\mathcal{T}^0 = v_0$, $\mathcal{V}^0_{sel} = [v_0]$. And in the k-th iteration, we first prompt the LLM to generate the k-th layer of the taxonomy, and update taxonomy $\mathcal{T}^{k-1}$ to $\mathcal{T}^k$, and remaining entity list $\mathcal{V}^{k-1}$ to $\mathcal{V}^k$. The k-th inference process in CoL is donated as:

$$\mathcal{T}^k, \mathcal{V}^k_{sel} = \text{CoL} - \text{K}(\mathcal{V}^{k-1}, \mathcal{T}^{k-1}, \mathcal{D}, \mathcal{R}) \tag{3}$$

$$\mathcal{V}^k = \mathcal{V}^{k-1} \setminus \mathcal{V}^k_{sel} \tag{4}$$

To alleviate the impact of the model's hallucinations on the quality of the output taxonomy, we employ the Ensemble-based Ranking Filter to filter out the hallucinations in the generated parent-child relations which are in $(\mathcal{T}^k \setminus \mathcal{T}^{k-1})$ at k-th iteration. Then we update the output taxonomy $\mathcal{T}^k$ and the remaining entity list $\mathcal{V}^k$ at k-th iteration. The processing of the Ensemble-based Ranking Filter is donated as:

$$\mathcal{T}^k, \mathcal{V}^k \leftarrow \text{EnsembleFilter}(\mathcal{T}^k, \mathcal{V}^k, \mathcal{V}^k_{sel}) \tag{5}$$

At the end of each iteration, we prompt the model to check if the remaining entity list $\mathcal{V}^k$ is empty or not. If we receive a positive response, we then output the current taxonomy $\mathcal{T}^k$ as the final result $\mathcal{T}$. Otherwise, we proceed to the next iteration.

	#Concepts	#Edges	Depth
WordNet	20.5 / 20.5	19.5 / 19.5	3.0 / 3.0
Wiki	102.6 / 252.0	101.8 / 255.0	2.2 / 3.0
DBLP	90.8 / 176.0	89.8 /175.0	2.8 / 4.0
SemEval-Sci	114.0 / 429.0	113.0 / 451.0	7.2 / 8.0

Table 1: Statistics of four taxonomy datasets. Each cell is presented as */*, indicating the average for sampled sub-taxonomies and the entire taxonomy, respectively.

3.4 Demonstrations Generation via LLMs

While CoL is designed to induct taxonomy from given entity sets via the few-shot learning setting. In domains that lack well-inducted taxonomies, we propose a zero-shot CoL alternative CoL-Zero. The idea of CoL-Zero is utilizing LLMs to generate taxonomies instead of utilizing taxonomies annotated by human experts acts as demonstrations.

The details of CoL-Zero are as follows. We start with the root entity v_0 of the target taxonomy $\mathcal{T}$. We follow the instruction mentioned in section 3.1 to prompt LLM directly to generate the taxonomies $\mathcal{T}_{\boldsymbol{d}_i^{(g)}}$, used to construct demonstration $\boldsymbol{d}_i^{(g)}$ following the process in section 3.2. Thus we have $\mathcal{D}^{(g)} = [\boldsymbol{d}_1^{(g)}, \cdots, \boldsymbol{d}_5^{(g)}]$. Different from CoL, CoL-Zero remove the restriction of only using the entities that are covered in the given entity set (r_1) to free form $\mathcal{R}' = [r_2, r_3]$.

$$\mathcal{T}_{\boldsymbol{d}_i^{(g)}} = \text{CoL} - \text{demo} - \text{generation}(\mathcal{T}^0, \mathcal{R}') \tag{6}$$

$$\mathcal{T} = \text{CoL}(\mathcal{V}, \mathcal{T}^0, \mathcal{D}^{(g)}, \mathcal{R}') \tag{7}$$

4 Experiments

Our proposed HF and CoL are evaluated on four benchmarks. The experiments aim to address three research questions (RQs):

- **RQ1:** How does the performance of the proposed framework compare to state-of-the-art baselines in taxonomy induction?
- **RQ2:** How does the proposed framework perform on scalability and domain generalization?
- **RQ3:** Which components within the proposed framework most significantly impact the effectiveness of taxonomy induction tasks? How can the hyperparameters for these components be determined?

4.1 Experimental Setting

4.1.1 Datasets. We conducted our experiments using WordNet sub-taxonomies created by [2]. This dataset comprises 761 non-overlapping taxonomies, each with 11 to 50 entities and the depth of each sub-taxonomy is 4. It means there are 4 entities along the longest path from the root entity to any leaf entity. The WordNet is divided into training (533), development (114), and test (114) sets.

Furthermore, we evaluate our framework using three large-scale real-world taxonomies: (1) **DBLP** is constructed from 156,000 computer science paper abstracts; (2) **Wiki** is derived from a subset of English Wikipedia pages; (3) **SemEval2016-Sci** is derived from the shared task of taxonomy induction in SemEval2016. For **DBLP** and **Wiki**, we uses annotation results from [27].

Model	WordNet					
	$\mathbf{P}_a$	$\mathbf{R}_a$	$\mathbf{F1}_a$	$\mathbf{P}_e$	$\mathbf{R}_e$	$\mathbf{F1}_e$
Supervised Fine-tuning						
Graph2Taxo [26]	79.20	47.80	59.60	75.60	37.00	49.70
CTP [6]	69.30	66.20	66.70	53.30	49.80	51.50
CTP-Llama-2-7B [6]	73.48	70.02	71.71	55.42	51.98	53.64
Zero-shot Setting						
RestrictMLM [13]	23.23	25.69	24.09	24.17	25.65	24.89
LMScorer [13]	37.50	47.64	41.59	36.27	38.48	37.34
Ours						
HF (GPT-4)	81.13	**78.35**	**78.37**	53.27	54.63	53.87
HF (GPT-3.5)	85.92	61.75	69.62	45.91	43.57	44.15
CoL-zero (GPT-4)	**89.71**	71.39	78.31	**58.93**	**55.18**	**56.41**
CoL-zero (GPT-3.5)	86.92	60.06	69.61	48.78	42.04	44.50
5-shot Setting						
TaxonomyGPT (GPT-4) [5]	53.09	31.84	39.07	39.59	36.84	38.01
TaxonomyGPT (GPT-3.5) [5]	62.97	41.77	48.95	49.20	43.85	46.24
Ours						
HF (GPT-4)	85.33	**79.30**	**81.58**	58.96	**59.22**	**59.08**
HF (GPT-3.5)	80.48	72.59	75.37	49.95	49.26	49.46
CoL (GPT-4)	**90.60**	73.07	79.62	**59.57**	57.10	57.73
CoL (GPT-3.5)	85.69	60.16	69.39	47.90	41.92	44.26

Table 2: Performance comparison across WordNet sub-taxonomies in three different settings: Bold indicates the highest performance within each setting, while underlined denotes the second best performance within each setting.

Due to the sequence length limitation of LLMs, we conducted five separate samplings for these three large-scale taxonomies, ensuring that the size of each sampled sub-taxonomy ranged from 80 to 120 entities. The experimental results are averaged over these five samplings. The dataset statistics are presented in Table 1.

4.1.2 Baseline Methods. We compare the proposed framework with the following supervised fine-tuning baseline methods:

- **Graph2Taxo** [26]: leverages cross-domain graph structures and adopts constraint-based Directed Acyclic Graph (DAG) learning for taxonomy induction.
- **CTP** [6]: fine-tunes RoBERTa model to predict parent-child pair likelihoods and integrates these into a graph using a maximum spanning tree algorithm for precise taxonomy induction. Additionally, we present results using a Llama-2-7B model as the backbone for CTP.

We compare the following unsupervised and in-context learning baseline methods:

- **RestrictMLM** [13]: utilizes a cloze statement, or 'fill-in-the-blank', method to extract 'is-a' relational knowledge from BERT. However, this approach is limited to single-gram entities due to the constraints of the schema.
- **LMScore** [13]: treats taxonomy induction as a sentence scoring task using GPT-2. It assesses the natural fluency of sentences that elicit parent-child relations.
- **TaxonomyGPT** [5]: approaches taxonomy induction as a conditional text generation challenge. It represents the output taxonomy as a collection of sentences, each describing a parent-child relation within the output taxonomy.

For our proposed framework, we conduct experiments with GPT-3.5-turbo-16k and GPT-4-1106-preview. For HF, we directly prompt the LLMs using the HF instruct describe in Section 3.1.

Model	Wiki						DBLP						SemEval-Sci					
	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$
Supervised Fine-tuning																		
Graph2Taxo [26]	43.02	36.50	39.49	39.28	34.12	36.52	47.85	30.23	37.05	46.63	28.49	35.37	82.45	36.15	50.27	79.37	34.52	46.87
CTP [6]	50.94	47.15	48.97	46.56	42.53	44.45	45.62	41.39	43.40	38.21	33.73	35.83	52.41	33.88	41.16	31.18	29.42	30.27
CTP-Llama-2-7B [6]	67.74	64.16	65.78	63.64	60.07	61.80	48.73	39.88	43.86	44.39	35.81	39.64	61.98	54.09	57.77	48.33	41.92	44.90
Zero-shot Setting																		
RestrictMLM [13]	49.88	54.08	51.85	30.01	30.21	30.11	-	-	-	-	-	-	63.33	47.85	54.44	45.79	46.19	45.99
LMScorer [13]	18.77	25.94	21.74	19.78	19.95	19.86	17.14	21.54	19.04	25.84	26.12	25.98	48.80	33.24	39.51	42.20	42.58	42.39
Ours																		
HF (GPT-4)	92.96	**94.48**	**93.68**	91.55	**91.31**	**91.41**	52.70	**64.69**	**57.65**	30.76	29.58	29.91	78.56	**54.68**	**64.02**	45.12	**46.64**	45.85
HF (GPT-3.5)	75.85	71.55	73.36	71.67	73.63	72.03	50.20	48.28	48.76	27.66	26.51	26.98	70.25	40.91	51.17	28.69	28.11	28.29
CoL-zero (GPT-4)	**100.00**	84.58	91.12	**99.70**	84.77	91.15	**80.88**	54.25	57.21	40.15	**35.88**	37.81	**94.99**	45.83	61.66	**62.33**	45.55	**52.44**
CoL-zero (GPT-3.5)	99.72	57.92	72.65	99.17	58.23	72.76	76.78	38.39	49.36	**53.72**	30.61	**38.02**	93.12	22.43	35.59	56.52	22.13	31.54
5-shot Setting																		
TaxonomyGPT [5]	69.26	63.48	65.19	89.55	86.71	87.98	28.98	14.40	17.15	34.27	22.17	25.97	53.09	31.84	39.07	39.59	36.84	38.01
Ours																		
HF-Zero (GPT-4)	96.33	95.18	95.75	93.08	91.72	92.39	59.76	**74.37**	65.83	38.42	40.20	39.28	75.28	**59.32**	62.63	43.64	**49.29**	45.24
HF-Zero (GPT-3.5)	88.88	80.36	84.38	83.67	74.92	78.98	62.42	53.76	57.53	32.68	28.59	30.38	57.00	36.89	44.38	29.51	29.88	29.35
CoL (GPT-4)	**99.17**	**95.99**	**97.54**	**97.92**	**94.99**	**96.43**	**79.95**	63.06	**68.82**	55.07	**44.27**	**47.96**	91.23	48.16	**62.69**	59.60	46.03	**51.59**
CoL (GPT-3.5)	99.00	73.25	83.73	97.54	71.99	82.41	79.74	42.21	54.76	**55.35**	28.70	37.66	**95.75**	26.66	41.35	**59.73**	26.05	35.99

Table 3: Performance on taxonomy induction on three large scale taxonomies: Bold for the highest among all. Underlined for the second-best performance. Due to the scalability challenges discussed in Section 4.4, each method was applied to five sub-taxonomies derived from the original, with results averaged. The RestrictMLM results for DBLP are unavailable since it only handles single-gram entities using a 'fill-in-the-blanks' schema. Due to GPT-4 not following the instructions of the TaxonomyGPT's prompt, only the results from TaxonomyGPT (GPT-3.5) were retained.

4.1.3 Evaluation Metrics. This section outlines the metrics for evaluating our taxonomy prediction models: Ancestor-F1 and Edge-F1.

Ancestor-F1: This metric assesses ancestor-descendant relations in predicted and ground truth taxonomies.

$$P_a = \frac{\left|\text{is-ancestor}_{\text{pred}} \bigcap \text{is-ancestor}_{\text{gold}}\right|}{\left|\text{is-ancestor}_{\text{pred}}\right|}$$

$$R_a = \frac{\left|\text{is-ancestor}_{\text{pred}} \bigcap \text{is-ancestor}_{\text{gold}}\right|}{\left|\text{is-ancestor}_{\text{gold}}\right|}$$

$$F1_a = \frac{2P_a * R_a}{P_a + R_a}$$

where P_a, R_a and $F1_a$ donate the ancestor precision, recall, and F1-score, respectively.

Edge-F1: This metric, stricter than Ancestor-F1, compares predicted edges directly with gold standard edges. Edge-based metrics are denoted as P_e, R_e, and $F1_e$, respectively.

4.2 Results on the WordNet (RQ1)

In our experiments, we compare the performance of our HF and CoL to three major settings baseline methods (supervised fine-tuning, zero-shot setting, and 5-shot setting) on medium-sized WordNet. As the experimental results are shown in Table 2, we have four major observations as follows:

Firstly, HF (GPT-4) and CoL (GPT-4) variants consistently achieved the highest F1 scores, validating the effectiveness of GPT-4 models in taxonomy induction. They significantly outperformed the LMScorer baseline, highlighting the superior text understanding and generation capabilities of LLMs.

Second, despite using powerful LLMs like GPT-4, TaxonomyGPT performed worse than methods such as CTP, which rely on fine-tuning BERT/Llama-2-7B models, across all six metrics. This suggests that LLMs are sensitive to output format requirements. TaxonomyGPT's approach of representing parent-child relationships as independent sentences loses structural coherence. In contrast, HF and CoL use a hierarchical number format to encode positional information, improving performance.

Third, Graph2Taxo achieved the highest precision across all settings, leveraging lexical patterns as direct input features. However, its lower recall indicates a trade-off, suggesting it may not fully capture all taxonomic relations.

Last, comparing HF and CoL, we have the following observations: (1) CoL-Zero (GPT-4) outperforms HF-Zero (GPT-4) with a 9.6%, 1.0%, and 4.5% increase in P_e, R_e, and $F1_e$. This result demonstrates that CoL is better suited for medium-sized taxonomy induction tasks under the zero-shot setting. (2) Under the 5-shot setting, HF (GPT-4) shows a 2.3% lead in $F1_e$ compared to CoL (GPT-4), indicating that direct prompting with HF achieves state-of-the-art results when in-domain examples are provided.

4.3 Results on the Three Large-Scale Taxonomies (RQ1 and RQ2)

In this section, we present the experiment results on three large-scale taxonomies: Wiki, DBLP, and SemEval-Sci, as shown in Table 3. This experiment tests domain generalization ability, with all supervised fine-tuning trained on WordNet and then tested directly on these taxonomies. Under the 5-shot setting, we use the first five sub-taxonomies of the WordNet training set for a fair comparison. Our observations are as follows:

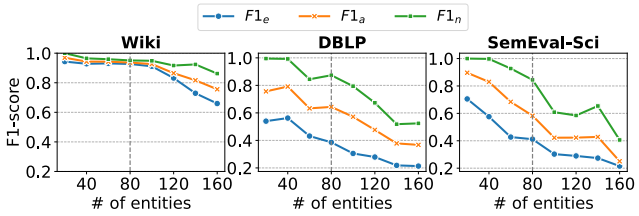


Figure 5: Performance analysis of the CoL across varying scales and domains. It shows Edge, Ancestor, and Node F1-scores for Wiki, DBLP, and SemEval-Sci taxonomies, ranging from 20 to 160 entities. An inflection point at the 80-entity threshold across all metrics and domains, emphasizing the scalability limitations of CoL.

First, in the supervised fine-tuning setting, models such as CTP and Graph2Taxo provide a foundation for understanding taxonomy induction's intricacies. However, the proposed CoL (GPT-4) shows the best performance in $F1_e$ across all three taxonomies. Compared to the best-performing SFT model, CoL (GPT-4) has increased $F1_e$ by 56.03%, 20.99%, and 10.07% on Wiki, DBLP and SemEval-Sci, respectively. Compared to HF, CoL (GPT-4) has increased $F1_e$ by 4.37%, 22.09% ,and 14.04% on Wiki, DBLP and SemEval-Sci, respectively.

Second, CoL-Zero demonstrates stronger domain adaptation capabilities than HF-Zero under the zero-shot setting. In the context of zero-shot learning, CoL (GPT-4) shows a 26.41% improvement on $F1_e$ than HF (GPT-4) in DBLP, and for SemEval-Sci, CoL (GPT-4) achieves 14.42% improvement than GPT-4. Although HF-Zero shows a better $F1_e$ than CoL-Zero, the increase in HF-Zero's $F1_e$ over CoL-Zero is only 0.285%, indicating that this marginal improvement is insufficient to prove that HF-Zero has superior domain adaptation capability compared to CoL-Zero.

These observations can be attributed to two reasons: (1) CoL decomposes taxonomy induction into different sub-tasks, such as focusing on finding parent-child relationships within a given layer, which enables the model to learn how to do taxonomy induction domain transfer across different domains. (2) The Ensemble-based Ranking Filter effectively improves the model's precision without sacrificing recall. Compared to using the proposed filter to post-process the output results once, CoL allows the generated results to be corrected by the proposed filter at every iteration of building the taxonomy. This mechanism enables the model to perform self-correction on the output taxonomy based on the existing context.

4.4 Investigating the Effects of Scalability on the Chain-of-Layer (RQ2)

In this section, we empirically investigate the scalability of our proposed Chain-of-Layer framework across varying scales (number of entities in the given entity list), with particular emphasis on identifying a critical threshold below which proposed CoL demonstrates optimal performance. We conduct experiments on Wiki, DBLP, and SemEval-Sci taxonomies. For each taxonomy, we randomly select sub-taxonomies, using the root entity as the starting point. We chose sub-taxonomies of various sizes, specifically with 20, 40, ..., 140, and 160 entities. To ensure the reliability of our results, we repeated the sampling process five times for each size,

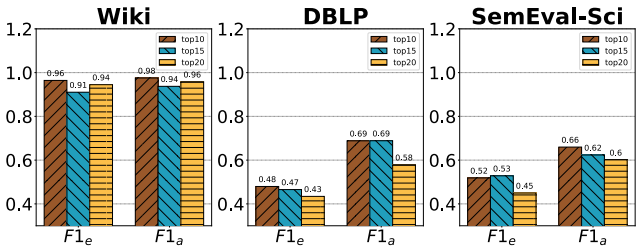


Figure 6: Performance of ranking ranges in the Ensemble-based Ranking Filter for maintaining parent-child relationships. The Top-10 range shows the highest $F1_a$ scores across all datasets and the highest $F1_e$ scores in Wiki and DBLP. In SemEval-Sci, the Top-10 range achieves nearly the best $F1_e$ score, close to the Top-15 range.

thereby generating five distinct sub-taxonomies for every specified number of entities. We not only report the trend of edge-level F1-score ($F1_e$ and $F1_a$) as it changes with variations in the size of the sub-taxonomy but also explore the trend of node-level F1-score ($F1_n$) on each dataset in Figure 5. Our observations are as follows.

First, as the scale of the taxonomy to be induced expands, both the edge-level F1-score and node-level F1-score of the proposed CoL framework exhibit a decline across all three benchmarks that in different domains. This correlation demonstrates that in the approaches that rely on prompting large language models, the increase in the number of entities significantly increases the complexity of the taxonomy induction task, leading to a relative performance decline even though the target taxonomy is in the same domain.

Secondly, in comparison to DBLP and SemEval-Sci, CoL exhibits robustness on the Wiki taxonomy. Specifically, even when expanding the entity count in the taxonomy to 160, CoL on Wiki shows a decrease in $F1_e$ and $F1_a$ of 30.01% and 22%, respectively, compared to when the entity count is 20. In contrast, on DBLP, $F1_e$ and $F1_a$ decrease by 60.50% and 51.41%, respectively, and SemEval-Sci, $F1_e$ and $F1_a$ decrease by 69.49% and 72.02%, respectively. This differential performance decline indicates that LLMs have a stronger knowledge understanding in general domains than in specific domains, such as the scientific domain.

Last, we find that the node-level F1-score ($F1_n$) also decreases more drastically as the number of entities exceeds 80 on DBLP and SemEval-Sci. Notably, the $F1_n$ remains relatively high with 20-80 entities, it sharply declines beyond this point. These findings indicate that when the taxonomy scale exceeds a certain threshold (beyond 80 entities), LLMs struggle to strictly adhere to the rules mentioned in the instructions: *using only the entities provided in the given entity set to carry out taxonomy induction.* This is also one of the significant reasons for the substantial decrease in the performance of CoL as the taxonomy scale increases.

4.5 Investigating the Effects of Hyperparameters & Ablation Study (RQ3)

4.5.1 The selection of the best ranking range. To identify the best ranking range for maintaining the parent-child relationships produced by CoL, we evaluated the top-10, top-15, and top-20 rankings across three large-scale taxonomies. Given that the top-1 and top-5 rankings scored below 50%, we consider them too stringent to accurately preserve the correct parent-child relationships. The results

Dataset	Configuration		Edge			Ancestor		
	CoL	Filter	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$
WordNet	X	✓	60.67	47.76	51.77	84.22	56.52	64.72
	✓	X	59.12	58.41	58.76	90.11	74.31	80.77
	✓	✓	59.57	57.10	57.73	90.60	73.07	79.62
Wiki	X	✓	98.08	46.27	61.09	99.49	45.58	60.66
	✓	X	98.58	93.08	95.73	99.51	93.96	96.63
	✓	✓	97.92	94.99	96.43	99.17	95.99	97.54
DBLP	X	✓	72.81	13.35	22.41	71.92	7.66	13.70
	✓	X	48.47	38.87	42.14	69.29	59.94	63.44
	✓	✓	55.07	44.27	47.96	79.95	63.06	68.82
SemEval-Sci	X	✓	57.29	20.10	29.32	93.33	13.91	23.94
	✓	X	54.22	49.75	51.86	86.29	53.74	65.97
	✓	✓	59.60	46.03	51.59	91.23	48.16	62.69

Table 4: Ablation study of two major modules in the proposed framework: Chain-of-Layer prompting (CoL) and Ensemble-based Ranking Filter (Filter). All metrics are presented in percentages (%). Configurations indicate whether CoL and the Ensemble-based Ranking Filter were employed.

are presented in Figure 6. The results are illustrated in Figure 6. Our findings reveal that for both $F1_e$ and $F1_a$, the top-10 ranking consistently demonstrates the best or second-best performance across all three datasets. Consequently, we selected the top-10 ranking as the optimal range for preserving the parent-child relationship in the Ensemble-based Ranking Filter.

4.5.2 Ablation Study. We conducted an ablation study on the four benchmarks mentioned above to verify the effectiveness of two major modules: Chain-of-Layer prompting (CoL) and the Ensemble-based Ranking Filter (Filter) in the proposed framework. The experimental results are shown in Table 4. Our findings are as follows:

First, removing CoL or Ensemble-based Ranking Filter reduces performance on three three large-scale taxonomies (Wiki, DBLP, and SemEval-Sci). It proves that the incorporation of CoL and Ensemble-based Ranking Filter provide crucial self-correction, reducing hallucinated content.

Second, the most notable drop in recall and F1-score performance occurs when the CoL is removed. It indicates that utilizing the Ensemble-based Ranking Filter as a post-processing iteration for the generated taxonomy proves overly stringent in maintaining the parent-child relations, even when those relations are correct. On the DBLP dataset, the absence of CoL results in a decrease of 65.7% in R_e and 46.8% in $F1_e$, despite a 50.2% improvement in P_e.

Third, removing the Ensemble-based Ranking Filter results in a decline in precision performance across all four benchmarks. This indicates that the proposed filter effectively preserves the accuracy of the parent-child relationship within the generated taxonomy.

Last, the introduction of CoL and the Ensemble-based Ranking Filter does not significantly impact the performance on WordNet. It is because WordNet's smaller scale allows models like GPT-4 Turbo to handle the task effectively without these enhancements.

4.6 Case Study

This section presents a case study to evaluate the strengths and weaknesses of our proposed methods alongside several baselines. We use samples from WordNet and provide outputs for CoL, CoL-w/o-Filter, HF (GPT-4), and HF (GPT-4)-Filter in a 5-shot setting.

4.6.1 CoL v.s. HF (GPT-4) / HF-Filter (GPT-4). By comparing the outputs of the CoL in Figure 7(b) and HF (GPT-4) in Figure 7(c)) / HF (GPT-4)-Filter in Figure 7(d), we demonstrate the effectiveness and importance of our CoL framework and the Ensemble-based Ranking Filter. Compared to the ground truth, the most noticeable issue with HF (GPT-4)'s output is that it hallucinates an entity *knife* that wasn't in the given entity list and uses it to group all other entities that should belong to *table knife*. This resulted in a lower edge F1 score for the generated taxonomy.

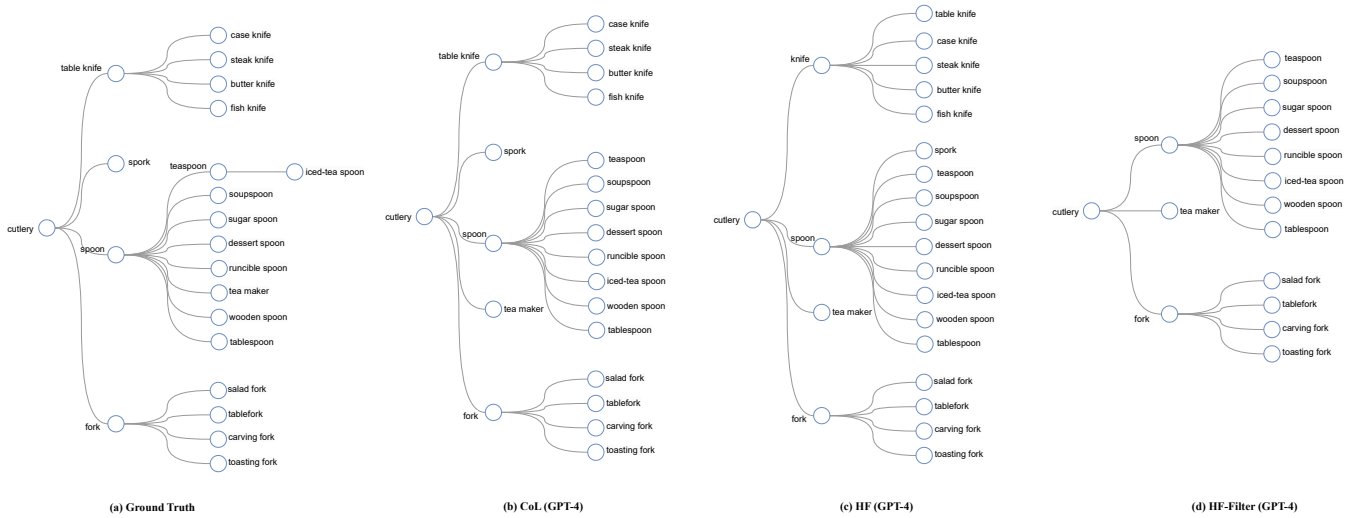
When comparing the outputs of HF (GPT-4) and HF (GPT-4)-Filter, we observe that without a layer-by-layer decomposition approach like CoL, directly employing a filter degrades the quality of the induced taxonomy. Filtering HF (GPT-4)'s output results in the complete removal of the sub-taxonomy under *knife*. This significantly lowers the node F1 score and edge F1 score of the generated taxonomy because filtered entities cannot be re-selected. These findings highlight the critical importance and synergistic effect of CoL and the Ensemble-based Ranking Filter.

4.6.2 CoL v.s. CoL-w/o-Filter. To illustrate the role of the Ensemble-based Ranking Filter, we compare the outputs of CoL in Figure 8(b) and CoL-w/o-Filter in Figure 8(c) against the ground truth in Figure 8(a). As shown, the taxonomy induced by CoL closely aligns with the ground truth, whereas CoL-w/o-Filter misclassifies "*roll*", "*bank*", and "*loop*" as siblings of "*flight maneuvers*". This demonstrates the effectiveness of the Ensemble-based Ranking Filter, which removes edges with lower ranks, such as "*flight maneuvers - roll*" and re-adds "*roll*" for selection in the next layer. This self-correcting process helps LLMs induce more accurate taxonomies.

5 Related Works

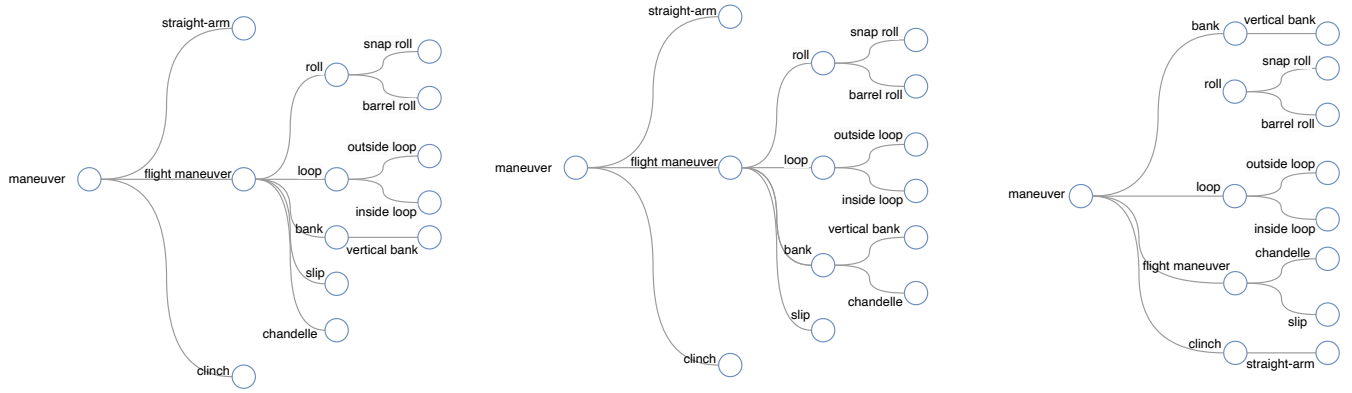
Taxonomy Induction. The process of taxonomy induction typically includes identifying hypernyms (extracting potential parent-child relationships from text) and and organizing them hierarchically. In the initial stage, embedding-based approaches [20] and pattern-based methods [22, 29, 33] were widely used. The second step was often viewed as graph optimization and solved by maximum spanning tree [2]. In this work, we focus on organizing a given entity set to a taxonomy. Bansal *et al.* [2] approach this as a structured learning problem and employ belief propagation to integrate relational information between siblings. Mao *et al.* [21] present an approach utilizing reinforcement learning to integrate the phases of hypernym identification and hypernym organization. Shang *et al.* [26] utilize a graph neural network approach, demonstrating improvement in large-scale taxonomy induction using the SemEval-2016 Task 13 dataset [4]. Chen *et al.* [6] and Jain *et al.* [13] utilize the pre-trained language model to approach taxonomy induction, treating it as sequence classification and sequence scoring tasks, respectively. Langlais and Guo [18] proposed an automatic taxonomy evaluation metric based on the pre-trained model. TaxonomyGPT [5] conducts taxonomy induction by leveraging the in-context learning capabilities of LLMs. The proposed CoL in this paper significantly reduces hallucination and improves structural accuracy by iterative prompting large language models.

Extracting Knowledge from LLMs. Research in extracting and investigating the stored knowledge in Large Language Models



(a) Ground Truth (b) CoL (GPT-4) (c) HF (GPT-4) (d) HF-Filter (GPT-4)

Figure 7: The taxonomies generated via CoL, HF (GPT-4) and HF-Filter (GPT-4).



(a) Ground Truth (b) CoL (c) CoL-w/o-Filter

Figure 8: The taxonomies generated via CoL and CoL-w/o-Filter

(LLMs) has become increasingly sophisticated, combining quantitative assessments with innovative extraction techniques. Prior works like LAMA [25], TempLAMA [8], and MMLU [11] have laid the groundwork by quantitatively measuring the factual and time-related knowledge within these models. Based on these, recent efforts have ventured into knowledge extraction, as seen in works that construct Knowledge Graphs (KGs) directly from LLM outputs. Specifically, methodologies like the one introduced in Crawling Robots [7] propose the extraction of named entities and relationships through a novel robot role-play setting, indicating a shift towards more interactive and dynamic extraction methodologies. Parallel to this, the adoption of structured, prompt-based queries has offered a pathway to not only retrieve but also systematically organize the knowledge embedded within LLMs, making it accessible and interpretable for human users [19, 37]. This emerging body of work, including techniques that enhance training data with explicit knowledge recitation tasks [30], aims at not just understanding but also effectively leveraging the vast reservoir of information encapsulated in these advanced models, marking a significant leap forward in our quest to harness the full potential of LLMs for knowledge-based applications.

6 Conclusion

In this work, we introduce Chain-of-Layer (CoL), a novel framework for taxonomy induction. By leveraging the hierarchical format instruction (HF) and incorporating an Ensemble-based Ranking Filter, CoL breaks down the task into selecting relevant candidates and gradually building the taxonomy from top to bottom and significantly reduces hallucination and improves structural accuracy. Extensive experimental results demonstrate that CoL outperforms various baselines, achieving state-of-the-art performance. We envision CoL as a powerful framework to address the challenges of inducting accurate and coherent taxonomy from a set of entities.

Acknowledgments

This work was supported by NSF IIS-2119531, IIS-2137396, IIS-2142827, IIS-2234058, CCF-1901059, and ONR N00014-22-1-2507.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1041–1051.

[3] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3615–3620.

[4] Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*. 1081–1091.

[5] Boqi Chen, Fandi Yi, and Dániel Varró. 2023. Prompting or Fine-tuning? A Comparative Study of Large Language Models for Taxonomy Construction. In *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 588–596.

[6] Catherine Chen, Kevin Lin, and Dan Klein. 2021. Constructing Taxonomies from Pretrained Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4687–4700.

[7] Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. 2023. Crawling The Internal Knowledge-Base of Language Models. In *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics, Dubrovnik, Croatia, 1856–1869. https://aclanthology.org/2023.findings-eacl.139

[8] Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-Aware Language Models as Temporal Knowledge Bases. *Transactions of the Association for Computational Linguistics* 10 (2022), 257–273. https://doi.org/10.1162/tacl_a_00459

[9] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1199–1209.

[10] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1709–1719.

[11] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*. https://openreview.net/forum?id=d7KBjmI3GmQ

[12] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 573–581.

[13] Devansh Jain and Luis Espinosa Anke. 2022. Distilling Hypernymy Relations from Language Models: On the Effectiveness of Zero-Shot Taxonomy Induction. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*. 151–156.

[14] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361* (2020).

[16] Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. TXtract: Taxonomy-Aware Knowledge Extraction for Thousands of Product Categories. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8489–8502.

[17] Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 conference on empirical methods in natural language processing*. 1110–1118.

[18] Philippe Langlais and Tianjian Lucas Gao. 2023. RaTE: a Reproducible automatic Taxonomy Evaluation by Filling the Gap. In *Proceedings of the 15th International Conference on Computational Semantics*. 173–182.

[19] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 3154–3169. https://doi.org/10.18653/v1/2022.acl-long.225

[20] Anh Tuan Luu, Yi Tay, Siu Cheung Hui, and See Kiong Ng. 2016. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 403–413.

[21] Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-End Reinforcement Learning for Automatic Taxonomy Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2462–2472.

[22] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1135–1145.

[23] 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. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[24] 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 *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 1320–1327.

[25] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2463–2473. https://doi.org/10.18653/v1/D19-1250

[26] 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 *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*. 2198–2208.

[27] 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 *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*. Springer, 288–304.

[28] Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2389–2398.

[29] Rion Snow, Daniel Jurafsky, and Andrew Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in neural information processing systems* 17 (2004).

[30] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-Augmented Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=-cqvvvb-NkI

[31] Yanchao Tan, Carl Yang, Xiangyu Wei, Chaochao Chen, Longfei Li, and Xiaolin Zheng. 2022. Enhancing recommendation with automated tag taxonomy construction in hyperbolic space. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1180–1192.

[32] Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39, 3 (2013), 665–707.

[33] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. 481–492.

[34] Yuejia Xiang, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Zhenxi Lin, and Yefeng Zheng. 2021. OntoEA: Ontology-guided Entity Alignment via Joint Knowledge Graph Embedding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 1117–1128.

[35] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large Language Models for Generative Information Extraction: A Survey. *arXiv preprint arXiv:2312.17617* (2023).

[36] Shuo Yang, Lei Zou, Zhongyuan Wang, Jun Yan, and Ji-Rong Wen. 2017. Efficiently answering technical questions—a knowledge graph approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[37] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than Retrieve: Large Language Models are Strong Context Generators. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=fB0hRu9GZUS

[38] Qingkai Zeng, Mengxia Yu, Wenhao Yu, Jinjun Xiong, Yiyu Shi, and Meng Jiang. 2019. Faceted hierarchy: A new graph type to organize scientific concepts and a construction method. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*.

[39] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. 2014. Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 243–252.