Rank, Chunk, and Expand: Lineage-Oriented Reasoning for Taxonomy Expansion

Sahil Mishra [♣,*] **Kumar Arjun** [♢] **Tanmoy Chakraborty** [♣]

[♣]Department of Electrical Engineering, IIT Delhi [♢]Department of Mathematics, IIT Delhi

[♣]`{sahil.mishra, tanchak}@ee.iitd.ac.in`, [♢]`kumar.arjun.mt121@maths.iitd.ac.in`

Abstract

Taxonomies are hierarchical knowledge graphs crucial for search engines, recommendation systems, and web applications. As data grows, expanding taxonomies is essential, but existing methods face key challenges: (1) discriminative models struggle with representation limits and generalization, while (2) generative methods either process all candidates at once, introducing noise and exceeding context limits, or discard relevant entities by selecting a sample of candidates. We propose `LORex` (**L**ineage-**O**riented **Re**asoning for Taxonomy E**x**pansion), a plug-and-play framework that combines discriminative ranking and generative reasoning for efficient taxonomy expansion without fine-tuning. Unlike prior methods, `LORex` ranks and chunks candidate terms into batches, filtering noise and iteratively refining selections by reasoning candidates' hierarchy to ensure contextual efficiency. Extensive experiments across four benchmarks and twelve baselines show that `LORex` improves accuracy by 12% and Wu & Palmer similarity 5% over state-of-the-art methods.

1 Introduction

Taxonomies are hierarchical graph structures that capture hypernymy ("*is-a*") relationships among concepts, making them essential for knowledge organization across various domains. Conglomerates leverage them to power search engines (Strzelecki and Rutecka, 2019; Janssen and Proper, 2021), enhance product recommendations (Mao et al., 2020; Karamanolakis et al., 2020) and improve advertisements (Gonçalves et al., 2019; Manzoor et al., 2020). Despite their utility, real-world taxonomies are manually created by domain experts, which is cumbersome and costly, limiting their ability to capture emerging concepts (Jurgens and Pilehvar, 2016; Bordea et al., 2016a). Additionally, as new concepts continuously emerge, updating taxonomies manually becomes increasingly impractical.

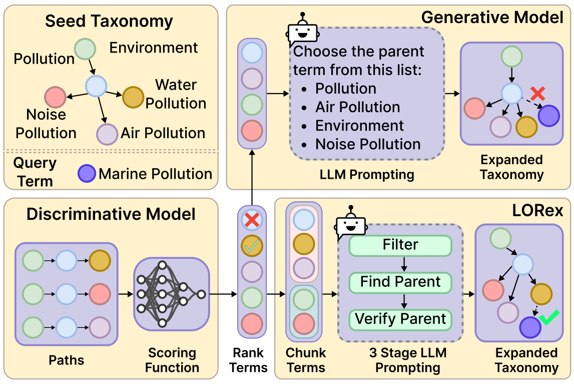

Figure 1: Illustration of taxonomy expansion task and contribution of `LORex` framework.

To overcome the limitations of manual taxonomy construction, research has now shifted toward taxonomy expansion, which involves the integration of new entities into an existing seed taxonomy (Jurgens and Pilehvar, 2016; Bordea et al., 2016a). This process positions new concepts under appropriate existing nodes, known as *anchor nodes*, to preserve the structural integrity of the original taxonomy. As illustrated in Fig. 1, this ensures that new terms, like "Marine Pollution", are inserted under "Water Pollution" rather than the broader "Pollution", maintaining hierarchical coherence.

Research on taxonomy expansion has evolved through three distinct generations. The first-generation approaches primarily rely on the seed taxonomy as weak supervision to assess hypernymy relationships, leveraging lexical pattern matching (Jurgens and Pilehvar, 2015) and distributional embeddings (Chang et al., 2018). However, these methods are inherently constrained by limited self-supervised annotation data and, therefore, fail to fully harness the hierarchical and structural intricacies embedded within taxonomies.

[*]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 12935–12953
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Second-generation research improves taxonomy expansion by incorporating structural summaries like mini-paths (Yu et al., 2020; Jiang et al., 2022), paths (Liu et al., 2021), random walks (Xu et al., 2022), ego-nets (Shen et al., 2020), and local graphs (Wang et al., 2021) to enrich hierarchical representation. These methods are also supplemented with external knowledge, such as domain-specific corpora (Yu et al., 2020) and concept definitions (Liu et al., 2021). However, they struggle with small-scale taxonomies, where limited self-supervised data hinders model training and generalization, requiring more scalable approaches.

The current generation research on taxonomy expansion leverages LLMs to integrate new concepts into existing taxonomies. Models like LLaMA (Touvron et al., 2023; Dubey et al., 2024) and GPT-4 (Achiam et al., 2023), excel in structural knowledge comprehension (Wei et al., 2023; Liu et al., 2025). Notably, GPT-4, with 1.7 trillion parameters, generates parent terms directly (Zeng et al., 2024b,a), while LLaMA-2 (7B parameters) requires fine-tuning on self-supervised annotation data for taxonomy expansion (Mishra et al., 2024; Xu et al., 2025).

While effective, these approaches incur high computational costs. Fine-tuning LLaMA models requires substantial resources, whereas GPT-4 inference is expensive due to its massive parameter size and paid API-based access, further increasing costs and reducing flexibility for scaling taxonomy expansion. Additionally, LLM-driven methods face context size limitations – some attempt to encode all candidates into prompts (Mishra et al., 2024), which is impractical, while others rely on top-k selection (Zeng et al., 2024b), risking the omission of relevant terms.

To address limitations of existing taxonomy expansion methods, we propose **`LORex`** (**L**ineage-**O**riented **Re**asoning for Taxonomy E**x**pansion), a plug-and-play that integrates discriminative ranking and generative reasoning to systematically expand taxonomies without fine-tuning LLMs or discarding relevant candidates.

Inspired by Liu et al. (2021), `LORex` first ranks the candidate terms for an entity using a discriminative ranker, named `TEMPORA`, which leverages lineage structures to score and rank candidates. `TEMPORA` converts the lineage into a taxonomy path via an Euler tour, then verbalizes relationships by labeling edges as *"parent of"* and *"child of"*. Ranked candidates are then chunked into batches, where an LLM first filters relevant batches and performs hierarchical reasoning to select the best parent. To verify the correctness, we introduce an LLM-based path-scoring function that ranks candidates based on their lineage structure.

We conduct extensive experiments to evaluate `TEMPORA` for candidate ranking and `LORex` for taxonomy expansion on four public benchmark datasets, including SemEval-2016 Task 13 Taxonomy Extraction Evaluation (Bordea et al., 2016a) and WordNet (Bansal et al., 2014). `TEMPORA` outperforms all ranking baselines, improving Hit@k by 21.3%, despite their extensive self-supervised training. `LORex` surpasses eight baselines across four benchmarks, achieving 12% higher accuracy and a 5% gain in Wu & Palmer (Wu&P) metric. Furthermore, we perform ablations to analyze the performance of `LORex` under varying conditions, including its effectiveness with different open-source LLMs.

We summarize our main contributions below:[1]

- We introduce `LORex`, a plug-and-play framework that combines discriminative and generative methods to rank and chunk candidates, followed by iterative reasoning over hierarchy to retrieve and verify parent terms for taxonomy expansion without fine-tuning LLMs or discarding relevant candidates.
- We develop `TEMPORA`, a discriminative ranker, which transforms lineage into path by performing an Euler tour on it, followed by path verbalization, where taxonomy edges are annotated with relational labels such as "parent of" and "child of" to enhance interpretability.
- Experiments on four benchmark datasets demonstrate that `TEMPORA` surpasses all ranking baselines with a 21.3% improvement in Hit@k, while `LORex` achieves a 12% increase in accuracy and a 5% boost in Wu&P metric, establishing an efficient state-of-the-art in taxonomy expansion.

2 Related Work

Taxonomy Expansion. Expert-curated taxonomies, such as The Plant List (Schellenberger Costa et al., 2023) and NCBI Taxonomy (Schoch et al., 2020), require continuous updates, necessitating robust taxonomy expansion. Recent discriminative methods leverage structural features such as taxonomy paths (Liu et al., 2021), mini-paths (Yu et al., 2020), hierarchical neighbour-

[1] Code: https://github.com/sahilmishra0012/LOREx.

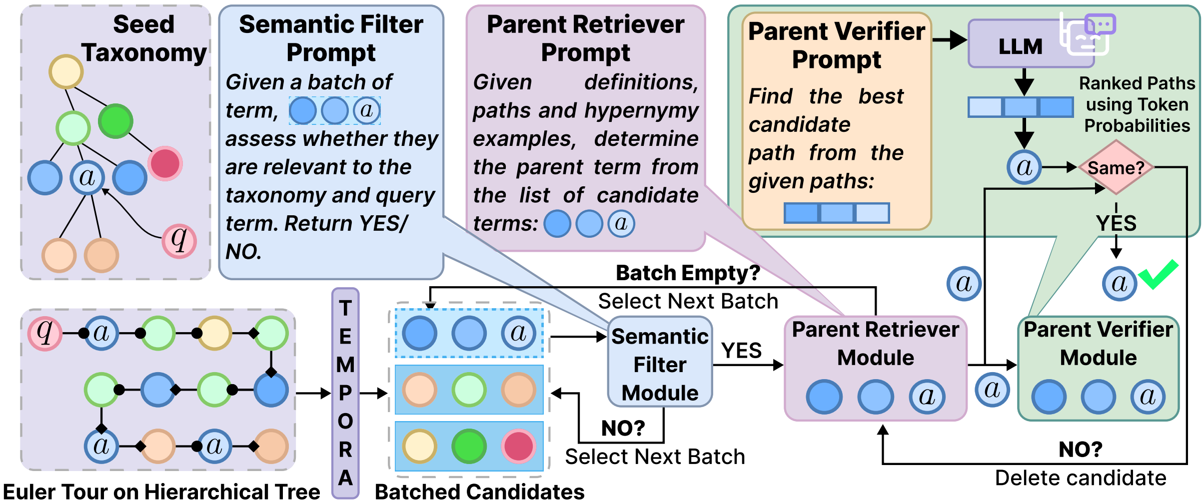

Figure 2: An illustration of our proposed framework, LORex, which first ranks candidates using TEMPORA and chunks them. The semantic filter module then checks the batch relevance, which further leads to parent retrieval and verification. In Euler tour, • (⧫) means "*is child of*" ("*is parent of*") relationship.

hood(Jiang et al., 2022), ego-trees (Wang et al., 2021), ego networks (Shen et al., 2020) and even quantum entanglements (Mishra et al., 2025), to enhance anchor node representation and model hypernymy relationships. Prompting frameworks such as TaxoPrompt (Xu et al., 2022), ATTEMPT (Xia et al., 2023), and TacoPrompt (Xu et al., 2023) model taxonomy paths via masked language modeling. However, these methods struggle with limited self-supervised annotated data and small-scale encoders. To address this, LLM-based solutions such as FLAME (Mishra et al., 2024) and COMI (Xu et al., 2025) fine-tune LLaMA models, while CodeTaxo (Zeng et al., 2024b) employs GPT-4 inference. Though effective, fine-tuning is computationally intensive, and GPT-4 API usage is costly. To address these challenges, our proposed LORex model integrates discriminative ranking with generative reasoning, leveraging smaller language models. LORex employs iterative prompting to systematically reason over taxonomic lineages, ensuring computational efficiency while maintaining high accuracy in taxonomy expansion.

Structure Reasoning using LLMs. Recent studies on structure reasoning with LLMs have explored graph-based integration to enhance knowledge grounding and reasoning capabilities. Studies such as Think-on-Graph (Sun et al., 2024) employs iterative beam searches over knowledge graphs (KGs) to retrieve reasoning paths, while Graph-CoT (Jin et al., 2024) extends Chain-of-Thought (CoT) prompting by incorporating graph structures into iterative reasoning cycles. Similarly, Paths-over-Graph (Tan et al., 2024) enhances reasoning over structure by integrating dynamic multi-hop path exploration and pruning techniques within KGs. Additionally, code language prompting methods like CodeKGC (Bi et al., 2024) and Code4struct (Wang et al., 2022) have also been used to reason over graphs by converting graph entities into classes following object-oriented programming paradigm. Building on these advances, our work focuses on iterative reasoning to enhance structural representation learning.

3 Problem Definition

Definition 1 ***Taxonomy:*** *A taxonomy* $\mathcal{T}^o = (\mathcal{N}^o, \mathcal{E}^o)$ *is a hierarchically structured directed acyclic graph (DAG), where each node* $n \in \mathcal{N}^o$ *denotes a distinct entity encapsulated by a concept and supplemented with a description* $D_n \in D$*, while each edge* $\langle n_p, n_c \rangle \in \mathcal{E}^o$ *signifies a "parent-child" dependency.*

As new entities continue to emerge, the key challenge is incorporating them into an existing taxonomy $\mathcal{T}^o$, referred to as taxonomy expansion,

Definition 2 ***Taxonomy Expansion:*** *Taxonomy expansion is the process of incorporating emergent concepts* C *into a seed taxonomy* $\mathcal{T}^o = (\mathcal{N}^o, \mathcal{E}^o)$ *to create an updated taxonomy* $\mathcal{T} = (\mathcal{N}^o \cup \mathcal{C}, \mathcal{E}^o \cup \mathcal{R})$*, where* $\mathcal{R}$ *is the set of new relations connecting existing entities* $\mathcal{E}^o$ *with the new concepts* C*.*

4 Proposed Method

LORex, a plug-and-play taxonomy expansion framework (Fig. 2), integrates discriminative ranking (TEMPORA) with iterative structure reasoning to integrate new concepts into an existing taxonomy.

4.1 TEMPORA

The first stage of LORex is to rank and chunk all candidate nodes $\mathcal{N}^o$ for all query nodes $\mathcal{C}$. Since LLMs have context length limitations, incorporating all candidate terms $\mathcal{N}^o$ in the prompt is impractical and expands the search space. Therefore, we need to rank candidates to prioritize relevance. We propose TEMPORA, a discriminative ranker (inspired by TEMP (Liu et al., 2021)), which uses pre-trained encoders to learn path-based features. Unlike TEMP, TEMPORA: (i) incorporates siblings and children to better capture hierarchical structures, and (ii) verbalizes paths for enhanced interpretability, whereas TEMP relies on special tokens like [SEP] or [UNK].

To effectively capture the hierarchical structure, we extract the path from the root node to the anchor node, along with its children and siblings. This hierarchy is then verbalized by performing an Euler tour on the tree, which begins at the anchor node, ascends to the root, and subsequently traverses siblings, followed by child nodes in a structured manner. The Euler-toured path is represented as $P = [n_p, n_{p-1}, \ldots, n_o, n_1, \ldots, n_{p-1}, n_{s_1}, \ldots, n_{p-1}, n_p, n_{c_1}, n_p, n_{c_2}]$, where n_o is the root node while n_{p-i}, n_{s_i} and n_{c_i} are the ancestor siblings and children of anchor node, respectively. Unlike traditional Euler tour, where each node can only be visited once, our version of Euler tour can visit a node more than once. Contrary to TEMP, we use interpretable relational phrases such as *"is parent of"* and *"is child of"* to enhance the clarity and interpretability of the Euler-toured path, returning verbalized path as P_v. We return two different verbalized paths $P_v(q,P) = \mathit{Verbalizer}(q,P)$ and $P_v(P) = \mathit{Verbalizer}(P)$ where q is the query node whose definition is used by the verbalizer, which is concatenated with path P via the ([SEP]) token. The pre-trained encoder returns the following vectors,

$$f(P_v(q,P)) = v_{\texttt{[CLS]}}, v_1, \ldots, v_{\texttt{[SEP]}}, v_{n_p}, \ldots, v_{n_{c_2}}, \tag{1}$$

$$f(P_v(P)) = v_{\texttt{[CLS]}} v_{n_p}, \ldots, v_{n_{c_2}}, \tag{2}$$

where $v_{\texttt{[CLS]}}$ represents the vectorized embedding of [CLS] token, which is processed through a Multi-Layer Perceptron (MLP) to compute the fitting score. TEMP is optimized using a dynamic margin loss that differentiates between positive ($P_v^+(q,P)$) and negative ($P_v^-(q,P)$) paths. However, TEMP undergoes extensive training across multiple benchmarks, as its primary objective is to predict the most suitable candidate term. In contrast, TEMPORA is a simple retriever, which should not need exhaustive training. A minimal number of training epochs should be sufficient for its intended function.

However, limiting training to a few epochs introduces several challenges, as the model struggles to distinguish between positive and negative paths effectively. To mitigate this, we introduce a dual-path training strategy wherein the model is simultaneously trained on $P_v(q,P)$ and $P_v(P)$. Specifically, for positive samples, we minimize the margin loss between these paths, while for negative samples, we maximize it. The losses are defined as,

$$\mathcal{L}_{\Updownarrow} = \sum_{P,P'} \max(0, -f(P) + f(P') + \gamma(P,P')) \tag{3}$$

$$\mathcal{L}_{+} = \sum_{P,P_r} \left\| f(P) - f(P_r) \right\|^2 \tag{4}$$

$$\mathcal{L}_{-} = \sum_{P',P'_r} \max\left(0, f(P') - f(P'_r)\right), \tag{5}$$

where $P \in P_v(q,P)^+$ and $P' \in P_v(q,P)^-$ are positive and negative verbalized paths with query definitions while $P_r \in P_v(P)^+$ and $P'_r \in P_v(P)^-$ exclude definitions. Visualization of different types of paths is shown in Appendix A. The joint loss is represented as,

$$\mathcal{L} = \mathcal{L}_{\Updownarrow} + \lambda_1 \cdot \mathcal{L}_+ + \lambda_2 \cdot \mathcal{L}_-, \tag{6}$$

where λ_1 and λ_2 are hyperparameters controlling the importance of the path difference losses. To measure the semantics of taxonomy paths, we use the dynamic margin function proposed by TEMP,

$$\gamma(P,P') = \left(\frac{|P \cup P'|}{|P \cap P'|} - 1 \right) \cdot d, \tag{7}$$

where d is the margin adjustment parameter. Self-supervised data sampling and training is done in the same way as for TEMP. During ranking, we rank all candidate terms $n_p \in \mathcal{N}^o$ against all query terms $n_c \in \mathcal{C}$ by computing fitting score $f(P_v(n_q, P_p))$, where P_p is the Euler-toured path of candidate term

n_p. Then, we chunk ranked candidate terms in batches of k. For a query term n_c^i, the batches are $\mathcal{B}^i = \{B_1^i, B_2^i, \ldots, B_m^i\}$, where each batch B_j^i contains k elements (except possibly the last batch).

4.2 Semantic Filter Module

After obtaining a batch of candidate terms B_j^i for the query term n_c^i, it is crucial to assess their relevance to the taxonomy or the query term. The filtering step ensures that only contextually relevant batches are passed for further task, enhancing retrieval precision while reducing computational overhead. For instance, a batch containing terms such as ["environmental policy", "environmental degradation", "environmental protection"] exhibits strong semantic alignment with the "environment" taxonomy, particularly in relation to "quality of the environment." In contrast, a batch comprising ["bird", "ornithology", "genetics"] lacks the necessary contextual relevance, making it less suitable for parent retrieval.

To achieve this, we introduce a semantic filter module, which ensures that only semantically coherent candidate terms advance to subsequent retrieval and verification stages, effectively reducing processing overhead by discarding irrelevant batches. The filtering process is executed via Boolean inference, where the LLM evaluates the semantic alignment of a batch with the taxonomy or query term. The LLM is prompted with query term n_c^i along with candidate terms B_j^i and their definitions D_j^i to achieve fair semantic evaluation. If the LLM returns 'Yes,' the batch proceeds to subsequent processing; otherwise, it is discarded (see Appendix B for the prompt).

4.3 Parent Retriever Module

Once the candidate batch B_j^i successfully passes the semantic filtering step, we utilize the lineage of the candidate terms to determine the most suitable parent for the given query term n_c^i. This process involves reasoning over the hierarchy of the candidate term, as discussed in Section 4.1, which involves leveraging path to the root node, ego network and definition of the candidate, to infer the most granular candidate that directly subsumes the query term. The LLM is prompted to select the most suitable hypernym from the candidate list, returning `NOT FOUND` if no valid hypernym exists (see Appendix C for the prompt).

However, a key limitation of the retriever is that the LLM does not explicitly return `NOT FOUND` when no suitable hypernym exists. This occurs because the candidate terms are pre-ranked, leading to semantically similar terms being grouped within the same batch. As a result, the LLM often selects a term that is closely related but not the most appropriate hypernym, failing to discard incorrect candidates and move to the next batch for a better match. To address this, a verification step is needed to ensure the predicted hypernym aligns with the taxonomy structure.

4.4 Parent Verifier Module

In order to verify if the retrieved parent from LLM is the most appropriate one, existing works have employed discriminative verifiers (Zeng et al., 2024a). However, these verifiers are often limited by their representation power. Moreover, if we use path-based verifier similar to `TEMPORA`, the verifier is again biased towards the ranked candidates, creating a bottleneck in verification. To overcome this, we instead utilize LLMs to reason over candidate paths within a batch to identify the most plausible hierarchical structure. However, direct reasoning over paths introduces instability, particularly when using smaller instruction-tuned LLMs. These models frequently generate inconsistent or malformed outputs, omitting full paths or introducing noise due to complex reasoning steps. To mitigate this, we compute the average token log-probability for each candidate path in the prompt and use these scores to rank them. This approach offers a more stable and interpretable signal for selecting the best-reasoned path.

The LLM is prompted with the hierarchical path of retrieved candidate from parent retriever module along with the path of remaining candidates within the batch to select the most appropriate path (see Appendix D for the prompt). Then, average token log probabilities are computed for all the paths in the batch to rank them to eliminate the possibility of the verifier selecting longer paths. The highest-scored candidate path is computed as,

$$P_j^* = \arg\max_{P_i \in B_j^i} \frac{1}{n} \sum_{i=1}^{n} \log p(t_i | t_1, \ldots, t_{i-1}), \quad (8)$$

where P_j^* represents the selected path with the highest cumulative log probability score while t_i denotes the i-th token in the path.

If the retrieved candidate path aligns with the verifier's selection, it is retained as the most appropriate hypernym; otherwise, the candidate is

Method	Environment		Science	
	Hit@1	Hit@10	Hit@1	Hit@10
BM25	0.077	0.250	0.118	0.212
SBERT$_{COS}$	0.173	0.615	0.294	0.576
BERT$_{COS}$	0.115	0.385	0.106	0.329
SimCSE	0.120	0.450	0.177	0.468
TEMP	<u>0.403</u>	<u>0.654</u>	<u>0.459</u>	<u>0.612</u>
TEMPORA	**0.481**	**0.731**	**0.529**	**0.753**

Table 1: Performance Comparison of TEMPORA with discriminative ranking baselines. The state of the art is **bolded**, while the best baseline is <u>underlined</u>.

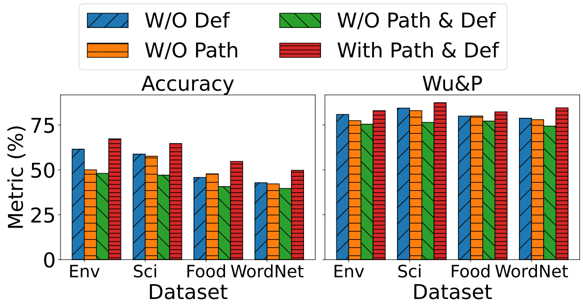

Figure 3: Performance comparison of hierarchical and semantic reasoning on parent retrieval across all benchmarks. Here W/O means **without**.

removed from the batch, and the parent retriever is re-invoked to select a new hypernym. This iterative process continues until only two candidates remain in the batch. If the verifier fails to correctly identify the retrieved path among the final two candidates, the entire batch is discarded, and the process moves to the next batch. The pseudocode of the framework is provided in Algorithm 1 in Appendix E.

5 Experiments

5.1 Experimental Setup

We discuss the evaluation of LORex and TEMPORA, covering benchmark datasets, baselines, and evaluation metrics. Implementation details are discussed in Appendix F.

Benchmark Datasets. We experiment on four open-source benchmarks: three from SemEval-2016 Task 13 Taxonomy Extraction Evaluation (Bordea et al., 2016b), Science (SemEval-Sci or Sci), Environment (SemEval-Env or Env) and Food (SemEval-Food) while the other is a collection of 114 WordNet sub-taxonomies (with 10 to 50 nodes each) (Bansal et al., 2014) (refer to Table 8 for statistics on benchmarks and Appendix G for train-test split and procedure to resolve definitions).

Baseline Methods. We select two parallel baseline methods, one for TEMPORA while another for LORex. We compare the discriminative ranker, TEMPORA, against commonly used ranking methods such as BM25, SimCSE and TEMP, Sentence-BERT, and BERT-Base-Uncased with cosine similarity. For LORex's evaluation, we use 12 baselines, namely, (i) **BERT+MLP** (Devlin et al., 2019), (ii) **TAXI** (Panchenko et al., 2016), (iii) **TaxoExpan** (Shen et al., 2020), (iv) **STEAM** (Yu et al., 2020), (v) **Musubu** (Takeoka et al., 2021) (vi) **TEMP** (Liu et al., 2021) (vii) **HEF** (Wang et al., 2021) (viii) **BoxTaxo** (Jiang et al., 2023), (ix) **TaxoPrompt** (Xu et al., 2022), (x) **TacoPrompt** (Xu et al., 2023) (xi) **TaxoComplete** (Arous et al., 2023), and (xii) FLAME (Mishra et al., 2024), which are discussed in Appendix H.

Evaluation Metrics. To evaluate the ranking performance of TEMPORA, we use Hit@k as an evaluation metric, which is the number of correctly predicted parents in the *top-k*. Since the ranked candidates are chunked, these metrics provide insight into whether the correct element appears within the *top-k* positions. For evaluating LORex, we follow the evaluation strategy used in CodeTaxo, which uses Accuracy and Wu & Palmer similarity (Wu&P) (discussed in Appendix I).

5.2 Performance Comparison

TEMPORA. As shown in Table 1, our TEMPORA model outperforms the best baseline, TEMP, across all Hit@k metrics on Env and Sci benchmarks (Refer to Table 9 in Appendix J for results on other benchmarks). Specifically, it achieves a 23.5% improvement in Hit@1 and a 19.1% increase in Hit@10. From this, we can conclude that path-based rankers outperform semantic rankers. The enhanced ranking accuracy directly translates to improved efficiency and reduced latency in both the retrieval and verification stages.

LORex. Table 2 shows the performance of LORex on four benchmarks. As discussed in Appendices C and D, our framework uses prompts in an instruction-tuned paradigm. Therefore, we use three instruction-tuned LLMs , LLaMA-3.1-8B-Instruct (LORex$_{8B-3.1I}$), LLaMA-3-8B-Instruct (LORex$_{8B-3I}$), and LLaMA-2-7B-Chat (LORex$_{7B-2C}$) to compare against state-of-the-art

Methods	SemEval16-Env		SemEval16-Sci		SemEval16-Food		WordNet	
	Acc	Wu&P	Acc	Wu&P	Acc	Wu&P	Acc	Wu&P
BERT+MLP	12.6 ± 1.1	48.3 ± 0.8	12.2 ± 1.7	45.1 ± 1.1	12.7 ± 1.8	49.1 ± 1.2	9.2 ± 1.2	43.5 ± 0.4
TAXI	18.5 ± 1.3	47.7 ± 0.4	13.8 ± 1.4	33.1 ± 0.7	20.9 ± 1.1	41.6 ± 0.2	11.5 ± 1.8	38.7 ± 0.7
Musubu	42.3 ± 3.2	64.4 ± 0.7	44.5 ± 2.3	75.2 ± 1.2	38.6 ± 2.7	63.4 ± 0.4	25.6 ± 4.7	61.4 ± 1.6
TaxoExpan	10.7 ± 4.1	48.5 ± 1.7	24.2 ± 5.4	55.6 ± 1.9	24.6 ± 4.7	52.6 ± 2.2	17.3 ± 3.5	57.6 ± 1.8
STEAM	34.1 ± 3.4	65.2 ± 1.4	34.8 ± 4.5	72.1 ± 1.7	31.8 ± 4.3	64.8 ± 1.2	21.4 ± 2.8	59.8 ± 1.3
TEMP	45.5 ± 8.6	77.3 ± 2.8	43.5 ± 7.8	76.3 ± 1.5	44.5 ± 0.3	77.2 ± 1.4	24.6 ± 5.1	61.2 ± 2.3
HEF	51.4 ± 2.8	71.4 ± 2.3	48.6 ± 5.3	72.8 ± 1.8	46.1 ± 4.3	73.5 ± 3.2	28.1 ± 4.4	64.5 ± 1.8
BoxTaxo	32.3 ± 5.8	73.1 ± 1.2	26.3 ± 4.5	61.6 ± 1.4	28.3 ± 5.1	64.7 ± 1.6	22.3 ± 3.1	58.7 ± 1.2
TaxoComplete	45.3 ± 1.7	63.4 ± 0.2	38.4 ± 0.7	54.8 ± 0.3	33.6 ± 2.5	56.1 ± 0.6	28.3 ± 3.9	56.2 ± 2.1
TacoPrompt	56.2 ± 3.2	82.1 ± 0.4	53.1 ± 6.6	76.3 ± 0.8	53.4 ± 3.7	<u>80.6 ± 0.4</u>	44.8 ± 4.1	72.3 ± 1.6
TaxoPrompt	51.9 ± 6.3	78.6 ± 1.7	58.3 ± 3.8	78.1 ± 0.7	49.5 ± 3.7	74.4 ± 1.4	38.5 ± 4.5	68.2 ± 1.2
FLAME	63.4 ± 1.9	**85.1 ± 0.3**	63.2 ± 4.1	<u>82.5 ± 1.2</u>	**58.7 ± 3.5**	78.1 ± 1.5	45.2 ± 1.3	71.5 ± 0.6
LORex$_{7B-2C}$	59.7 ± 1.3	78.5 ± 0.5	55.3 ± 2.5	76.4 ± 0.2	49.1 ± 2.2	78.3 ± 1.2	<u>49.1 ± 2.5</u>	<u>83.2 ± 0.2</u>
LORex$_{8B-3I}$	<u>65.1 ± 2.4</u>	78.3 ± 0.4	59.5 ± 2.1	79.3 ± 0.4	51.6 ± 2.2	79.6 ± 0.5	48.2 ± 1.7	82.9 ± 0.3
LORex$_{8B-3.1I}$	**67.3 ± 1.4**	<u>82.9 ± 0.0</u>	**64.7 ± 2.1**	**87.4 ± 0.7**	<u>55.3 ± 1.5</u>	**84.3 ± 0.5**	**49.5 ± 0.6**	**84.5 ± 0.1**

Table 2: Performance comparison of LORex against baseline methods. The best-performing method is **bolded**, while the best baseline is <u>underlined</u>. Results are reported as the average performance (± 1-std dev) over three runs, each using a different random seed, reported as a percentage (%).

Method	Env		Sci		WordNet	
	Acc	Wu&P	Acc	Wu&P	Acc	Wu&P
TEMPORA	67.31	82.99	64.71	87.42	49.78	84.58
Random Shuffle	5.76	47.64	2.35	48.95	1.51	38.95

Table 3: Effect of discriminative ranker on the performance of LORex across "Env", "Sci" and "WordNet" benchmarks.

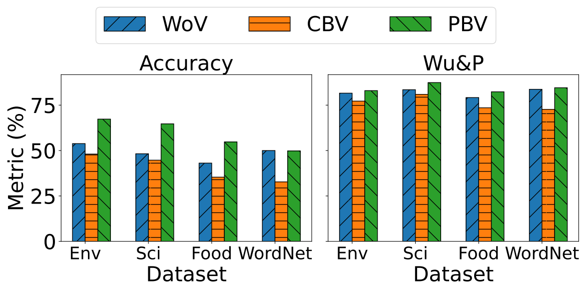

Figure 4: Performance comparison of different types of verifiers on SemEval-2016 benchmark. "WoV" = without verifier, "CBV" denotes candidate-based verifier, while "PBV" denotes path-based verifier.

methods. As shown, LORex surpasses all first and second-generation discriminative and third-generation prompting methods, with LORex$_{8B-3.1I}$ achieving the best performance. Specifically, LORex achieves 12% improvement in accuracy and 5% improvement in Wu&P over the best extensively trained generative baseline, FLAME.

5.3 Ablation Studies

We conduct ablations to analyze every component of LORex. Specifically, we analyze (i) the impact of hierarchical and semantic reasoning on parent retrieval, (ii) effect of different types of reasoning verifiers, (iii) the influence of discriminative ranker, (iv) varying chunk sizes on the overall performance of LORex, and (v) impact of semantic filter module on the performance and latency of LORex.

Impact of Hierarchical and Semantic Reasoning on Parent Retrieval. We analyze the impact of hierarchical and semantic reasoning by utilizing paths and definitions, on parent retrieval in LORex, as shown in Fig. 3. Using both paths and definitions gives the highest performance while removing both components leads to the worst performance, dropping accuracy by 28.3% and Wu&P by 5.3%. Removing definitions reduces accuracy by 17.50% and Wu&P by 2.78%. Further, removing hierarchical paths exacerbates the performance degradation, leading to a 31.99% decrease in accuracy and a 9.78% reduction in Wu&P similarity scores.

Effect of Different Types of Reasoning Verifiers on LORex's Performance. Fig. 4 compares LORex's performance under different verification strategies. The path-based verifier (PBV) achieves the highest accuracy, outperforming the candidate-

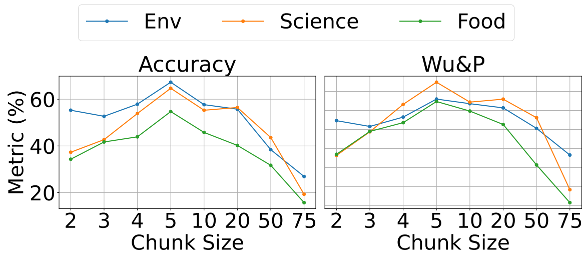

Figure 5: Effect of different chunk sizes on LORex's performance across the SemEval-2016 benchmarks.

Setting	Avg. Runs per Query	Queries Resolved in ≤3 Runs for "Env"
Semantic Filter	1.7	41 / 52
W/O Semantic Filter	3.1	18 / 52

Table 4: Impact of the semantic filter module on reasoning efficiency. W/O denotes without.

based verifier (CBV) [2] by 47.04% in accuracy and 10.84% in Wu&P scores. Even without a verifier (WoV), retrieval remains 21.31% more accurate and 7.76% higher in Wu&P than CBV. The lowest performance is observed with CBV, where path information is absent, and verification relies solely on candidate evaluation, which introduces noise during verification and rejects relevant candidates. These results emphasize the importance of hierarchical reasoning during verification.

Impact of Discriminative Ranker LORex's Performance. Table 3 compares LORex's performance with and without the use of a discriminative ranker. The discriminative ranker improves accuracy by 57.33% and Wu&P scores by 39.8%, highlighting its role in filtering relevant candidates and enhancing retrieval. In contrast, random shuffling of candidates disrupts the semantic ordering, often placing irrelevant or overly broad terms ahead of suitable hypernyms. This disordering hinders the retriever and verifier from progressing to later chunks where correct candidates may reside, thereby degrading overall system performance.

Effect of Different Chunk Sizes on LORex's performance. We analyze the impact of varying chunk sizes on LORex's retrieval performance across SemEval-2016 benchmarks, as shown in Fig. 5. The results indicate that smaller chunk sizes (2–3) lead to performance degradation due to limited contextual information, while excessively large chunk sizes (≥20) reduce effectiveness as the retriever encounters an increased number of irrelevant candidates, introducing noise in the chunk. The best performance is achieved with chunk sizes between 4 and 10, indicating a balance between relevant context depth and noise.

[2] CBV refers to a verifier setup where the LLM is directly prompted to rank candidate terms without access to their hierarchical paths. The ranking is based solely on average log token probabilities over candidate names, without structured reasoning over taxonomy paths.

Dataset	Batches with True Parent Skipped	Total Batches
Env	3	42
Sci	8	69
Food	13	238
WordNet	21	370

Table 5: Frequency of instances where the semantic filter skipped batches containing the true parent term.

Impact of Semantic Filter Module on Performance and Latency of LORex. As shown in Table 4, incorporating semantic filtering reduces the average number of retrieval-verification iterations per query from 3.1 to 1.7, and significantly increases the number of queries resolved within three runs. This demonstrates its utility in pruning irrelevant candidate batches early in the pipeline, reducing computational overhead and improving the latency of the framework. Furthermore, Table 5 shows that the semantic filter rarely eliminates batches containing the true parent node, only 3 out of 42 batches (for batch size=5) in the Environment dataset and similarly low rates across other datasets, indicating that the filter maintains high recall while enhancing precision. Failure cases of semantic filtering are discussed in Appendix K.

5.4 Case Studies

Table 6 discusses case studies for "Env" and "Sci" datasets, highlighting the reasoning mechanisms of LORex's components. We select two samples of each dataset and observe that parent retriever mostly reasons over paths to directly select the most appropriate parent from candidate terms as visible in case of query terms "*Arctic Ocean*" and "*exhaustion of resources*." However, it fails to retrieve the correct hypernym in case of "*pteridology*" whose definition does not clearly define the term, creating confusion for parent retriever. Moreover, the path-based verifier effectively exploits paths and reasons over them to distinguish between the retrieved path and alternatives, which is clearly evident in the case of "*pteridology*," where the best-

Query	Parent Retriever (Ground Truth)	Path-Based Verifier	Candidate-Based Verifier
Arctic Ocean *The Arctic Ocean is the smallest and shallowest of the world's oceans, located around the North Pole and mostly covered by sea ice.*	**ocean (ocean)** *The most appropriate direct hypernym of 'Arctic Ocean' is 'ocean.'*	**Arctic Ocean > ocean > geophysical environment > environment** *The given path is the most appropriate path for the query node 'Arctic Ocean'.*	**geophysical environment** *The query node 'Arctic Ocean' is a type of geophysical environment.*
exhaustion of resources *a resource is a source or supply from which a benefit is produced and that has some utility. resource exhaustion refers to the depletion of these resources due to overuse or unsustainable practices.*	**natural resources (natural resources)** *The most appropriate direct hypernym of 'exhaustion of resources' is 'natural resources.'*	**exhaustion of resources > natural resources > resource > possession** *The given path is 'exhaustion of resources -> natural resources -> resource -> possession', which appropriately contextualizes the query.*	**resource** *The query node 'exhaustion of resources' is a type of resource.*
pteridology *the scientific study of ferns and related plants known as pteridophytes. this field focuses on their classification, morphology, anatomy, ecology, distribution, and evolutionary history.*	**biology (botany)** *The most appropriate direct hypernym of 'pteridology' is 'biology.'*	**pteridology > botany > biology > life science > natural science > science** *The query node 'pteridology' is the scientific study of ferns. The most direct and relevant hypernym of 'pteridology' is 'botany', which is the scientific study of plants.*	**botany** *The query node 'pteridology' is the scientific study of ferns. The given candidate 'biology' is too broad as it encompasses the study of all living organisms. The alternative candidate 'botany' is more specific.*
realpolitik *realpolitik is a political approach that prioritizes practical considerations, and national interests over ideological concerns.*	**politics (politics)** *The most appropriate direct hypernym of 'realpolitik' is 'politics.'*	**realpolitik > politics > social science > science** *The given path is the most appropriate path for the query node 'realpolitik'.*	**social science** *The most appropriate candidate for the query node 'realpolitik' is 'social science'.*

Table 6: Case study analyzing the impact of LORex's components on the "Env" and "Sci" benchmarks. The table presents component outputs in **bold**, with their definitions and explanations in *italics*. Yellow highlighted text denotes key phrases in definitions and explanations that play an important role in reasoning.

inferred path is "*pteridology > botany > biology > life science > natural science > science*." In contrast, the candidate-based verifier relies only on definitions, which are noisy, incomplete and incorrect, to verify the retrieved candidate, as seen in the case of "*realpolitik*," where it fails to capture the appropriate hypernym, giving "*social science*" as the hypernym. These results emphasize the importance of path-based reasoning in verification after candidate retrieval over verification using definitions, which introduces unnecessary abstraction. More such cases are discussed in Appendix L.

6 Conclusion

We introduce LORex, a plug-and-play taxonomy expansion framework which integrates discriminative ranking and generative reasoning to effectively retrieve and verify parent-child relationships without requiring fine-tuning of large language models. By leveraging the discriminative ranker TEMPORA, the framework ranks candidate terms and optimally chunks them for LLM processing, ensuring relevant candidates are retained while maintaining context length constraints. Extensive experiments highlight that LORex outperforms the best baseline. Ablation studies highlight the importance of hierarchical reasoning, discriminative ranking, and verification strategies, which improve retrieval quality. Case studies further validate LORex's effectiveness in resolving ambiguous classifications and reinforcing hierarchical consistency.

Acknowledgements

We gratefully acknowledge the support of Microsoft through the Microsoft Academic Partnership Grant 2024 (MAPG'24). T. Chakraborty acknowledges the support of the Rajiv Khemani Young Faculty Chair Professorship in Artificial Intelligence.

Limitations

While LORex effectively retrieves parent terms in taxonomy expansion, its performance heavily relies on the quality of the candidate ranking. If the correct parent term is not highly ranked, LORex suffers from increased retrieval errors and higher computational overhead due to multiple iterations of retrieval and verification. This iterative process significantly impacts both latency and accuracy, particularly in cases where the ranker fails to surface relevant candidates early. As a result, the overall system efficiency is constrained by the discriminative ranker, making LORex highly sensitive to ranking quality. In other words, *"the method is only as good as its ranker."* Exploring better ranking techniques and reducing the reliance of retrievers on rankers will be the focus of our future works. We will also focus on improving interpretability by employing an ensemble of LLMs to reason over hierarchies. While this work focuses on taxonomy expansion, extending the framework to broader tasks such as taxonomy completion presents a promising direction for future research.

Ethical Considerations

Our study uses open-source taxonomies like WordNet, which may include words that are sensitive, controversial, or offensive. These words fall into different offensive categories, such as mental health terms, violent actions, political topics, religious groups, law enforcement agencies, and negative or insulting words. Some terms related to social and economic status, extreme beliefs, or reputation damage may also carry unintended meanings or reinforce biases. Since large language models can sometimes generate biased or inappropriate hypernyms, we take steps to make taxonomy expansion more responsible. To prevent misleading or harmful word relationships, we limit the model's hypernym choices to a predefined set from the seed taxonomies. This ensures that the generated hierarchies remain accurate and unbiased.

References

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*.

Ines Arous, Ljiljana Dolamic, and Philippe Cudré-Mauroux. 2023. Taxocomplete: Self-supervised taxonomy completion leveraging position-enhanced semantic matching. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2509–2518, New York, NY, USA. Association for Computing Machinery.

Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd ACL*, pages 1041–1051, Baltimore, Maryland. Association for Computational Linguistics.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2024. Codekgc: Code language model for generative knowledge graph construction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–16.

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

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016b. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *SemEval*, pages 1081–1091, San Diego, California. Association for Computational Linguistics.

Haw-Shiuan Chang, Ziyun Wang, Luke Vilnis, and Andrew McCallum. 2018. Distributional inclusion vector embedding for unsupervised hypernymy detection. In *NAACL*, pages 485–495, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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 *The Semantic Web – 18th ISWC 2019, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II*, page 418–435, Berlin, Heidelberg. Springer-Verlag.

Rik DT Janssen and Henderik A Proper. 2021. A functionality taxonomy for document search engines. *arXiv preprint arXiv:2105.12989*.

Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. 2022. Taxoenrich: Self-supervised taxonomy completion via structure-semantic representations. In *WWW*, page 925–934, New York, NY, USA. ACM.

Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. 2023. A single vector is not enough: Taxonomy expansion via box embeddings. In *WWW*, page 2467–2476, New York, NY, USA. Association for Computing Machinery.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, et al. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

David Jurgens and Mohammad Taher Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the WordNet taxonomy for novel terms. In *NAACL*, pages 1459–1465, Denver, Colorado. Association for Computational Linguistics.

David Jurgens and Mohammad Taher Pilehvar. 2016. Semeval-2016 task 14: Semantic taxonomy enrichment. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1092–1102.

Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. TXtract: Taxonomy-Aware Knowledge Extraction for Thousands of Product Categories. *Journal Name*, pages 8489–8502.

Ben Liu, Jihai Zhang, Fangquan Lin, Cheng Yang, and Min Peng. 2025. Filter-then-generate: Large language models with structure-text adapter for knowledge graph completion. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11181–11195, Abu Dhabi, UAE. Association for Computational Linguistics.

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*, pages 3854–3863, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *WWW*, page 2044–2054, New York, NY, USA. Association for Computing Machinery.

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 *SIGKDD*, page 2247–2257, New York, NY, USA.

Sahil Mishra, Avi Patni, Niladri Chatterjee, and Tanmoy Chakraborty. 2025. Quantaxo: A quantum approach to self-supervised taxonomy expansion. *arXiv preprint arXiv:2501.14011*.

Sahil Mishra, Ujjwal Sudev, and Tanmoy Chakraborty. 2024. Flame: Self-supervised low-resource taxonomy expansion using large language models. *ACM Trans. Intell. Syst. Technol.* Just Accepted.

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*, pages 1320–1327, San Diego, California. Association for Computational Linguistics.

David Schellenberger Costa, Gerhard Boehnisch, Martin Freiberg, Rafaël Govaerts, Matthias Grenié, Michael Hassler, Jens Kattge, Alexandra Muellner-Riehl, Blanca Rojas Andrés, Marten Winter, Mark Watson, Alexander Zizka, and Christian Wirth. 2023. The big four of plant taxonomy – a comparison of global checklists of vascular plant names. *New Phytologist*, 240.

Conrad L Schoch, Stacy Ciufo, Mikhail Domrachev, Carol L Hotton, Sivakumar Kannan, Rogneda Khovanskaya, Detlef Leipe, Richard Mcveigh, Kathleen O'Neill, Barbara Robbertse, Shobha Sharma, Vladimir Soussov, John P Sullivan, Lu Sun, Seán Turner, and Ilene Karsch-Mizrachi. 2020. Ncbi taxonomy: a comprehensive update on curation, resources and tools. *Database*, 2020:baaa062.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *WWW*, page 486–497, New York, NY, USA. Association for Computing Machinery.

Artur Strzelecki and Paulina Rutecka. 2019. The snippets taxonomy in web search engines. In *Perspectives in Business Informatics Research: 18th International Conference, BIR 2019, Katowice, Poland, September 23–25, 2019, Proceedings 18*, pages 177–188. Springer.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Kunihiro Takeoka, Kosuke Akimoto, and Masafumi Oyamada. 2021. Low-resource taxonomy enrichment with pretrained language models. In *EMNLP*, pages 2747–2758, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2024. Paths-over-graph: Knowledge graph empowered large language model reasoning. *arXiv preprint arXiv:2410.14211*.

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*.

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*, page 3291–3304, New York, NY, USA. Association for Computing Machinery.

Xingyao Wang, Sha Li, and Heng Ji. 2022. Code4struct: Code generation for few-shot event structure prediction. *arXiv preprint arXiv:2210.12810*.

Yanbin Wei, Qiushi Huang, Yu Zhang, and James Kwok. 2023. KICGPT: Large language model with knowledge in context for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8667–8683, Singapore. Association for Computational Linguistics.

Fei Xia, Yixuan Weng, Shizhu He, Kang Liu, and Jun Zhao. 2023. Find parent then label children: A two-stage taxonomy completion method with pre-trained language model. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1032–1042, Dubrovnik, Croatia. Association for Computational Linguistics.

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. IJCAI. Main Track.

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*, pages 15804–15817, Singapore. Association for Computational Linguistics.

Hongyuan Xu, Yuhang Niu, Yanlong Wen, and Xiaojie Yuan. 2025. Compress and mix: Advancing efficient taxonomy completion with large language models. In *THE WEB CONFERENCE 2025*.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *SIGKDD*, page 1026–1035, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 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*.

Appendix

A Modified Euler Tour

As discussed in Section 4.1, we present an example of modified Euler-toured path for taxonomy shown in Fig. 6. The Euler toured path is defined as P = ["water pollution", "pollution", "environment", "pollution", "air pollution", "pollution", "soil pollution", "pollution", "water pollution", "marine pollution", "water pollution", "chemical pollution"]. Therefore, the two types of verbalized paths (with and without query definition are discussed in Prompt 1 as shown in Fig. 8, where $P \in P_v(q, P)$ while $P_r \in P_v(P)$. These paths are used to compute three types of losses as shown in Fig. 7.

B Semantic Filter Prompt

The semantic filter module, discussed in Section 4.2, relies on LLM to check if the chunk of candidates is relevant by reasoning over their definitions. The prompt example is discussed in Prompt 2 shown in Fig. 9.

C Parent Retrieval Prompt

The parent retriever module, discussed in Section 4.3, reasons over taxonomy paths to determine the appropriate parent from the chunk of candidate terms. An example of the parent retriever prompt is discussed in Prompt 3 as shown in Fig. 10.

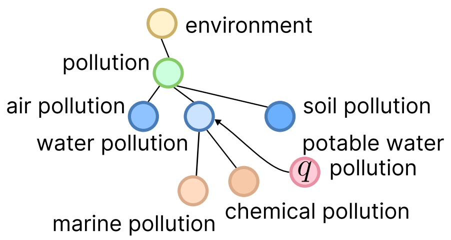

Figure 6: A hierarchical path of "water pollution" from "environment" taxonomy.

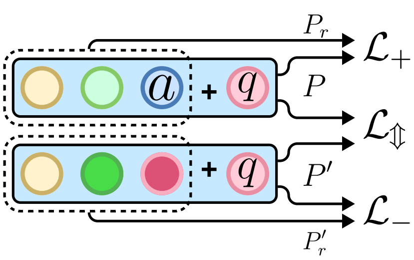

Figure 7: An illustration of loss functions defined in Section 4.1.

Hyperparameter	Sci	Food	Env	WordNet
Batch Size	32	32	64	32
Learning Rate	2e-5	1e-5	2e-5	1e-5
Max Padding	350	350	200	350
d	0.2	0.2	0.2	0.2
ϵ	1e-8	1e-8	1e-8	1e-8
Epochs	20	50	20	20

Table 7: Hyperparameter settings for different datasets.

D Parent Verifier Prompt

The parent verifier module verifies the appropriateness of the retrieved candidate by reasoning on paths as discussed in Section 4.4. An example of the prompt is discussed in Prompt 4 as shown in Fig. 11.

Algorithm 1: `LORex`

Data: candidate terms $\mathcal{N}^o$, query term n_c^i, large language model M, discriminative ranker `TEMPORA`
Result: candidate term n_p
`// Return sorted batches of candidates`
$\mathcal{B} \leftarrow$ `TEMPORA`$(n_c^i, \mathcal{N}^o)$;
`// Iterate through candidate batches`
for $B_j^i \in \mathcal{B}$ **do**
  if *Semantic-Filter*(B_j^i) **then**
    while $|B_j^i| > 1$ **do**
      $\hat{n}_p \leftarrow$ *Parent-Retriever*(q, B_j^i);
      `// Perform path verification and update batch`
      $P_j^* \leftarrow$ *Parent-Verifier*(q, B_j^i);
      if $\hat{n}_p \in P_j^*$ **then**
        return $\hat{n}_p$;
      else
        `// Remove retrieved candidate and continue`
        $B_j^i \leftarrow B_j^i \setminus \{\hat{n}_p\}$;
  $\mathcal{B} \leftarrow \mathcal{B} \setminus B_j^i$;
return *"Not Found"*

E Pseudocode

The pseudocode of `LORex` framework, which is discussed in Section 4, is shown in Algorithm 1.

Verbalized Paths

P

potable water pollution is the contamination of water that is intended for human consumption [SEP] "water pollution" is child of "pollution" is child of "environment" is parent of "pollution" is parent of "air pollution" is child of "pollution" is parent of "soil pollution" is child of "pollution" is parent of "water pollution" is parent of "marine pollution" is child of "water pollution" is parent of "chemical pollution"

P_r

"water pollution" is child of "pollution" is child of "environment" is parent of "pollution" is parent of "air pollution" is child of "pollution" is parent of "soil pollution" is child of "pollution" is parent of "water pollution" is parent of "marine pollution" is child of "water pollution" is parent of "chemical pollution"

Figure 8: Prompt 1 – Verbalized Path Examples of P and P_r.

For a query term n_c^i, the discriminator TEMPORA first ranks and chunks all candidate terms $\mathcal{N}^o$ as $\mathcal{B}$. Then, we select a chunk B and apply the parent retriever to get the most appropriate parent $\hat{n}_p$, followed by verification. If verification fails, the candidate is discarded, and the retrieval process is re-iterated on the remaining chunk. The iterative refinement continues until only one candidate remains in the chunk.

F Implementation Details

LORex and BERT+MLP are implemented in PyTorch, with all other baselines obtained from their respective official repositories. The training and inference processes are executed on a single 80GB NVIDIA A100 GPU to ensure computational efficiency. To implement TEMPORA, we use bert-base-uncased as the default pre-trained model, optimized using AdamW. The hyperparameters utilized for training and inference are detailed in Table 7. Notably, TEMPORA is trained on all benchmarks for fewer epochs compared to TEMP, as the primary objective is to achieve a ranker that delivers sufficiently high performance without excessive fine-tuning. Retrieval and verification rely on instruction-tuned models from the LLaMA family, namely LLaMA-3.1-8B-Instruct, LLaMA-3-8B-Instruct, and LLaMA-2-7B-Chat. As demonstrated in Fig. 12, all benchmarks except Food achieve a Hit@15 score above 0.9, indicating that for 90% of queries the correct parent exists within the top-15 candidates. Therefore, to enhance the efficiency of retrieval and verification, inference is performed on three chunks or 15 candidates simultaneously. If any chunk is empty, it is discarded, and only the remaining chunks are considered. This strategy ensures that the correct output is obtained in a single inference pass, optimizing both computational efficiency and retrieval accuracy. For text generation tasks, a stopping criterion is enforced to terminate generation upon encountering a newline character, which prevents the generation of unnecessary tokens, thereby reducing inference latency.

Dataset	Env	Sci	Food	WordNet
$\lvert\mathcal{N}^0\rvert$	261	429	1486	20.5
$\lvert\mathcal{E}^0\rvert$	261	452	1576	19.5
$\lvert D\rvert$	6	8	8	3

Table 8: Statistics of the four benchmark datasets. Here, $|\mathcal{N}^0|$ and $|\mathcal{E}^0|$ indicate the number of nodes and edges in the initial taxonomy, respectively, while $|D|$ represents the taxonomy depth. For WordNet, $|\mathcal{N}^0|$ and $|\mathcal{E}^0|$ denote the average number of nodes and edges across 114 sub-taxonomies.

Method	**Food**		**WordNet**	
	Hit@1	**Hit@10**	**Hit@1**	**Hit@10**
BM25	0.071	0.165	0.065	0.171
SBERT$_{COS}$	0.182	0.380	0.113	0.390
BERT$_{COS}$	0.077	0.118	0.076	0.225
SimCSE	0.119	0.280	0.258	0.621
TEMP	<u>0.330</u>	<u>0.529</u>	<u>0.329</u>	<u>0.682</u>
TEMPORA	**0.451**	**0.663**	**0.405**	**0.794**

Table 9: Comparison of TEMPORA with the discriminative ranking baseline methods. The best performance is marked in bold, while the best baseline is underlined.

G Benchmark Datasets

The statistics of benchmark datasets, as discussed in Section 5.1, namely Environment, Science, Food

You are a semantic relevance expert for terms present in a taxonomy. Your task is to determine whether the set of candidate terms is a semantically relevant match for the given query term 'Arctic Ocean' and 'Environment' taxonomy.

List of Candidate terms:
- geophysical environment
- ocean
- wild mammal
- animal life
- climatic zone

Definitions of candidate terms:
- geophysical environment - environment most often refers to:
- ocean - an ocean is a body of water that composes much of a planet's hydrosphere
- wild mammal - mammals, fur or hair, and three middle ear bones
- animal life - animals are multicellular eukaryotic organisms that form the biological kingdom animalia
- climatic zone - climate classification systems are ways of classifying the world's climates

Reason over the definitions of candidate terms to determine their relevance as a whole with respect to query terms Arctic Ocean and Environment taxonomy. If at least one of the candidate terms is the most relevant parent term for the taxonomy, return YES. Otherwise, return NO.

Answer:

Figure 9: Prompt 2 – Semantic Filter Prompt Example

Query	True Parent	Batch Skipped
fire protection	environmental protection	[**environmental protection**, environment, nuisance, environmental policy, pollution control measures]
desert	geophysical environment	[environment, degradation of the environment, **geophysical environment**, physical environment, wildlife]
demography	sociology	[social science, **sociology**, anthropology, economics, correlation]
thanatology	science	[endocrinology, anesthesiology, **science**, internal medicine, podiatry]
eugenics	genetics	[biology, **genetics**, life science, medical science, molecular biology]
scrumpy	cider	[**cider**, hard cider, sweet cider, alcohol, hooch]

Table 10: Failure cases where the LLM-based semantic filter incorrectly filters out batches with true parent.

and WordNet are shown in Table 8. For WordNet, the total number of nodes and edges are 2309 and 2226, respectively across 114 sub-taxonomies. We use GPT-4o to correct missing, mislabeled, or corrupted definitions. Following (Liu et al., 2021), we randomly select 20% of leaf concepts for testing.

H Baseline Methods

As mentioned in Section 5.1, we compare the performance of LORex against twelve baseline methods.

- **BERT+MLP** (Devlin et al., 2019) utilizes BERT to get term embeddings to predict hypernymy relationships.
- **TAXI** (Panchenko et al., 2016), a winner of SemEval-2016 Task 13 is a hypernym detection-based method that constructs taxonomies by identifying hypernymy relations between entity pairs based on lexical pattern extraction and substring matching.
- **TaxoExpan** (Shen et al., 2020) uses position-enhanced Graph Neural Networks (GNNs) to encode local structure. It models anchor representations using ego-network encoding and scores hypernymy relations with a log-bilinear feed-forward model, optimizing learning via the InfoNCE loss.
- **STEAM** (Yu et al., 2020) leverages mini-paths to model the local structure of the anchor nodes. It integrates graph-based, contextual, and manu-

You are an expert in hypernymy (is-a) relationship detection for a taxonomy. Your task is to find the most appropriate candidate hypernym of the query node 'Arctic Ocean' within the 'environment' taxonomy. The most appropriate hypernym is the most granular category that directly encompasses the query term.

List of Candidate terms:
- geophysical environment
- ocean
- wild mammal
- animal life
- climatic zone

Definitions of candidate terms:
- geophysical environment - environment most often refers to:
- ocean - an ocean is a body of water that composes much of a planet's hydrosphere
- wild mammal - mammals, fur or hair, and three middle ear bones
- animal life - animals are multicellular eukaryotic organisms that form the biological kingdom animalia
- climatic zone - climate classification systems are ways of classifying the world's climates

Path from query term to root node for all candidate terms:
- Arctic Ocean -> geophysical environment -> environment
- Arctic Ocean -> ocean -> geophysical environment -> environment
- Arctic Ocean -> wild mammal -> animal life -> wildlife -> environment
- Arctic Ocean -> animal life -> wildlife -> environment
- Arctic Ocean -> climatic zone -> climate -> environment

Query Node Definition: The Arctic Ocean is the smallest and shallowest of the world's oceans.

Some examples of hypernymy relationships in the taxonomy are as follows:
- Children of geophysical environment are: arid zone, estuary, island, lake, mountain, ocean, plain, polar region, sea
- Parent of animal life is: wildlife
- Children of climatic zone are: equatorial zone, frigid zone, humid zone, subtropical zone, temperate zone, tropical zone
- Parent of climatic zone is: climate

Instructions:
- Determine the most appropriate direct hypernym of 'Arctic Ocean' from the given list of candidate terms. Do not return any term which is not in the list.
- A hypernym must be a most granular category in which the query node is an instance.
- If no candidate term correctly fits as a hypernym , return NOT FOUND.
- Do not include explanations, justifications, or additional context.

Answer:

Figure 10: Prompt 3 – Parent Retriever Prompt Example

ally crafted lexical-syntactic features for query-anchor pairs and employs multi-view co-training to enhance hypernymy detection.

- **Musubu** (Takeoka et al., 2021) leverages pre-trained models, fine-tuning them to classify hypernymy. It utilizes queries derived from Hearst patterns to guide the fine-tuning process.
- **TEMP** (Liu et al., 2021) leverages a pre-trained model to encode definitions of taxonomy concepts while integrating structural information through taxonomic paths optimizing through dynamic margin loss.
- **HEF** (Wang et al., 2021) concatenates neighbor embeddings to construct the ego-tree representation and learns Stopper and Pathfinder scorers to identify candidate nodes in the taxonomy.
- **BoxTaxo** (Jiang et al., 2023) employs box embeddings to represent hierarchical relationships and optimizes geometric and probabilistic losses to score parent nodes. It leverages the volume of hyper-rectangles to model taxonomic structures effectively.
- **TaxoPrompt** (Xu et al., 2022) prompts encoder models such as BERT to capture taxonomic structures. It formulates prompt templates using random walks to represent taxonomy context and optimizes the model via the masked language modeling objective.
- **TacoPrompt** (Xu et al., 2023) employs multi-task learning to mitigate overfitting taxonomy

You are an expert verifier of hypernymy relationship for a taxonomy using paths. You have been given the following path from query term 'Arctic Ocean' to root node: Arctic Ocean -> ocean -> geophysical environment -> environment

Your task is to verify if the given path is the most appropriate path for the query node or the following paths provide a better alternative.

Other possible paths:
- Arctic Ocean -> geophysical environment -> environment
- Arctic Ocean -> ocean -> geophysical environment -> environment
- Arctic Ocean -> wild mammal -> animal life -> wildlife -> environment
- Arctic Ocean -> animal life -> wildlife -> environment
- Arctic Ocean -> climatic zone -> climate -> environment

Some examples of hypernymy relationships in the taxonomy are as follows:
- Children of geophysical environment are: arid zone, estuary, island, lake, mountain, ocean, plain, polar region, sea
- Parent of animal life is: wildlife
- Children of climatic zone are: equatorial zone, frigid zone, humid zone, subtropical zone, temperate zone, tropical zone
- Parent of climatic zone is: climate

Definitions of candidate terms:
- geophysical environment - environment most often refers to:
- ocean - an ocean is a body of water that composes much of a planet's hydrosphere
- wild mammal - mammals, fur or hair, and three middle ear bones
- animal life - animals are multicellular eukaryotic organisms that form the biological kingdom animalia
- climatic zone - climate classification systems are ways of classifying the world's climates

Query Node Definition: The Arctic Ocean is the smallest and shallowest of the world's oceans.

Instructions: - Compare the given path with the alternative paths based on definitions and hypernymy relationships.
- Select the most appropriate path that best represents the hierarchical relationship of 'Arctic Ocean' in the taxonomy.

Return the most appropriate path.

Answer:

Figure 11: Prompt 4 – Parent Verifier Prompt Example

completion tasks. It leverages prompt-based semantic triplet matching to enhance the model's generalization capabilities.

- **TaxoComplete** (Arous et al., 2023) integrates a query-anchor semantic matching mechanism with a direction-aware propagation module, facilitating a more accurate hierarchical representation of nodes.
- **FLAME** (Mishra et al., 2024) fine-tunes low-rank parameters of LLMs to predict parent terms by leveraging semantically relevant local and global neighborhood information of candidates.

We adapt taxonomy completion baselines, such as TaxoComplete and TacoPrompt for the taxonomy expansion setting by introducing a pseudo-placeholder in place of the child node. This transformation ensures that the task is framed as a leaf node insertion, allowing these completion methods, originally designed for intermediate node prediction, to be effectively applied to taxonomy expansion tasks.

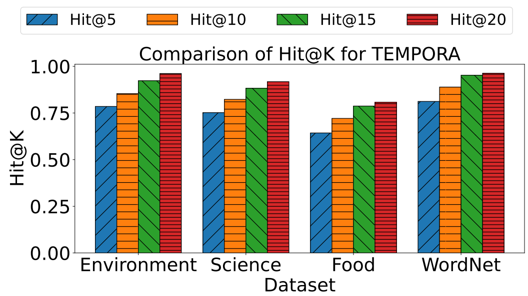

Figure 12: Comparison of Hit@K performance for `TEMPORA` across different benchmarks (Environment, Science, Food, and WordNet).

Query	Parent Retriever (Ground Truth)	Path-Based Verifier	Candidate-Based Verifier
grapefruit peel *grapefruit peel has a bright, citrusy flavor with a balance of bitterness and sweetness.*	**candied citrus peel (candied citrus peel)** *The most appropriate direct hypernym of ’grapefruit peel’ is ’candied citrus peel.’*	**grapefruit peel > candied citrus peel > candied fruit > confiture > sweet > dainty > nutriment > food** *The given path is the most appropriate path because it directly connects ’grapefruit peel’ to the root node ’food’ through a series of hypernymy relationships that are relevant to the definition of ’grapefruit peel’.*	**lemon peel** *The definition of lemon peel is more specific and closely related to the query node ’grapefruit peel’ compared to the other candidates.*
demography *the scientific study of human populations, including their size, composition, distribution, density, and growth.*	**sociology (sociology)** *The most appropriate direct hypernym of ’demography’ is ’sociology.’*	**demography > sociology > social science > science** *The path is appropriate because ’demography’ is a subfield of sociology, which belongs to social science, a domain within science. This path reflects a logical hierarchical classification of ’demography’.*	**social science** *Although social science is a broader category that includes demography, the term ’sociology’ is more specific and directly connected to the study of population structures. Therefore, ’sociology’ better captures the essence of the query than ’social science’.*
Irish soda bread *A variety of quick bread traditionally made in a variety of cuisines in which sodium bicarbonate is used as a leavening agent instead of the traditional yeast.*	**bread (quick bread)** *The most appropriate direct hypernym of ’Irish soda bread’ is ’bread.’*	**Irish soda bread > quick bread > bread > starches > foodstuff > food** *The query node ’irish soda bread’ is a type of ’quick bread’ as per the definition. ’Quick bread’ is a type of ’bread’ as per the hypernymy relationship. ’Bread’ is a type of ’starches’ as per the hypernymy relationship. ’Starches’ is a type of ’foodstuff’ as per the hypernymy relationship. ’Foodstuff’ is a type of ’food’ as per the hypernymy relationship.*	**quick bread** *The query node ’irish soda bread’ is a type of quick bread as it uses sodium bicarbonate as a leavening agent instead of traditional yeast.*
differential calculus *a branch of calculus concerned with the study of the rates at which quantities change.*	**calculus (calculus)** *The most appropriate direct hypernym of ’differential calculus’ is ’calculus’.*	**differential calculus > calculus > pure mathematics > mathematics > science** *The path traverses through calculus and correctly situates the concept in the broader scientific hierarchy.*	**calculus** *The candidate calculus is appropriate.*
destruction of crops *a crop is a plant or animal product that can be grown and harvested for profit or subsistence. Destruction refers to damage to this resource.*	**degradation of environment (degradation of environment)** *The most appropriate direct hypernym of ’destruction of crops’ is ’degradation’ of environment.*	**destruction of crops > degradation of the environment > environment** *This path is appropriate because degradation of the environment includes anthropogenic harm to agricultural systems.*	**degradation of environment** *The candidate node degradation of environment is valid.*

Table 11: Case study analyzing the impact of LORex’s components on "Env", "Food" and "Science" benchmarks.

I Evaluation Metrics

During inference, both the baselines and LORex rank all candidate terms for each query node. Given a query set $\mathcal{C}$, the predictions are denoted as $\{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{|\mathcal{C}|}\}$ while the corresponding ground truths are represented as $\{y_1, y_2, \cdots, y_{|\mathcal{C}|}\}$. Following CodeTaxo ((Zeng et al., 2024b)), we evaluate performance using two key metrics, as outlined in Section 5.1,

- **Accuracy (Acc):** This metric quantifies the proportion of predicted candidates that correctly match the ground-truth,

$$\text{Acc} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\mathbb{I}\left(y_i = \hat{y}_i\right), \tag{9}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.
- **Wu & Palmer Similarity (Wu&P):** This metric quantifies the structural similarity between the predicted parent and the ground truth within the seed taxonomy. It is computed based on the closest common ancestor,

$$\text{Wu\&P} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\frac{2 \times \text{depth}\left(\text{LCA}\left(\hat{a}_i, a_i\right)\right)}{\text{depth}\left(\hat{a}_i\right) + \text{depth}\left(a_i\right)}, \tag{10}$$

where depth$(\cdot)$ represents the depth of the node, and LCA$(\cdot,\cdot)$ denotes the least common ancestor of the predicted candidate and ground truth.

J Discriminator Results

Table 9 provides additional empirical evidence supporting the performance of TEMPORA, as analyzed in Section 5.2. The results indicate that TEMPORA consistently outperforms TEMP across both evaluation metrics on the Food and WordNet benchmarks.

K Failure Cases of Semantic Filtering

While the semantic filter module is effective in improving efficiency and recall (as discussed in Section 5.3), there are a few failure cases where it incorrectly filters out batches containing the true parent term. These errors typically stem from ambiguity or insufficient granularity in the definitions used for semantic matching. As illustrated in Table 10, for the query term *fire protection*, the filter erroneously rejects the batch containing the correct parent *environmental protection*, with the LLM reasoning that fire protection does not semantically fall under that category. Similar misclassifications are observed for terms like *desert* and *thanatology*, where broader parent categories such as *geophysical environment* and *science* were excluded due to narrow or domain-specific interpretations. These cases highlight the limitations of LLM-based semantic inference when reasoning over abstract or generalized definitions, though such instances remain infrequent across datasets.

L Case Studies

Extending on the case studies discussed in Section 5.4, we further analyze some additional examples with more semantically rich explanations in Table 11. The parent retriever provides straightforward reasoning over definitions and path-based relationships. However, as it is susceptible to errors, we rely on the verifier's reasoning to validate and refine the retriever's predictions. The path-based verifier employs lineage-oriented reasoning to assess retrieval accuracy by analyzing hierarchical paths. For instance, in the case of 'grapefruit peel', the verifier explains, "The given path is the most appropriate as it directly connects 'grapefruit peel' to the root node 'food' through a series of hypernymy relationships that align with its definition." In contrast, the candidate-based verifier relies on definitions, which can lead to incorrect predictions when contextual hierarchical relationships are not explicitly captured in definitions.