New Insights on Old Beliefs: Ontology-based Self Evolution of Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) excel in general domains but lack specialized knowledge. Existing methods use external annotated data to enhance LLMs, which is resource-intensive. We propose a novel framework for LLM's self-evolution in specialized domains using ontology-driven knowledge extraction and enhancement. We introduce BeliefConf, a metric to quantify the model's confidence in knowledge paths, and our method of the Automated Path Annotation Mechanism (APAM) helps identify Enhanced Paths for targeted training. Experiments show that our method outperforms the base model (Llama3-8B-instruct) on 3 out of 6 medical datasets (PubMedQA, MedQA, USMLE-step1) and achieves state-of-the-art performance on PubMedQA without external training data, surpassing models like Llama3-Med42-8B.

1 Introduction

Current Large Language Models (LLMs) have demonstrated remarkable capabilities in general domains(Wang et al., 2024). While these models also exhibit some proficiency in handling specialized questions, they still lack sufficient knowledge in professional domains(Ling et al., 2023).

Many existing methods enhance LLMs by injecting domain-specific knowledge (Christophe et al., 2024; Gururajan et al., 2024) into fine-tuning pretrained models. However, these approaches require extensive annotated data from specialized domains, which is both labor-intensive and resource-consuming.

In general domains, model's self-evolution has garnered growing attention from researchers lately due to its independence from external supervised data(Tao et al., 2024). Researchers have proposed various methods to make models generate and annotate training data autonomously based on specific priciples, such as consistency(Wang et al., 2023a; Madaan et al., 2023), multi-step reasoning(Yu et al., 2024), or ethical requirement integration(Sun et al., 2023). However, these principles and the their associated evolution objectives tend to be too generalized, making them unsuitable for highly specialized fields demanding rigorous professional knowledge and conceptual understanding.

Recent research (Liu et al., 2025) suggests that improving models' understanding of domain ontologies can greatly enhance their performance in specialized fields, which inspires us to consider the possibility of leveraging ontology to enable models' self-evolution in specialized domains. This is because Ontologies inherently incorporate built-in rules and inconsistency detection mechanisms, making them a powerful tool for structuring and validating domain-specific knowledge. If we can extract knowledge within models into an explict ontology, we can utilize these ontology rules to extrapolate on knowledge and detect inconsistencies within them. Furthermore, ontologies are rich in concepts and their interrelationships, with each knowledge path clearly represented in a triple format (subject-predicate-object). This structured representation allows for precise identification and enhancement of weak or incomplete domain knowledge in a point-to-point manner, addressing the specific evolutionary needs of models in specialized domains.

Specifically, in our method, we first extract model's internal knowledge in domains into an explicit ontology. We then automatically identify which parts of this ontology require updates using our **Automated Path Annotation Mechanism (APAM)**. APAM consists of two main steps. First, we annotate reliable paths within the extracted ontology, based on the assumption that a knowledge path is more likely to be reliable if a majority of its supporting paths exist within the model. During the process, We introduce a novel metric called **BeliefConf** to quantify model's confidence in each

knowledge path. Second, we infer new paths based on the reliable paths and verify whether these inferred paths are also recognized by the model. If not, we consider the situation as inconsistency and classify them as enhanced paths then generate targeted training corpora to improve the model's familiarity towards these paths.

We conducted experiments in the medical domain, using Llama3-8B-instruct(Dubey et al., 2024) as the base model and fine-tuning it with our proposed method. We compared our approach against the base model as well as other domain-specific models fine-tuned from Llama3-8B by external domain corpus. Results show that our method outperforms the base model on 3 out of 6 medical evaluation datasets, PubMedQA(Jin et al., 2019), MedQA(Jin et al., 2020), and USMLE-step1(Han et al., 2023), and significantly surpasses all baseline models on the PubMedQA dataset, including Llama3-Med42-8B(Christophe et al., 2024), which is fine-tuned on external data and achieves the best performance on the remaining datasets.

Our main contributions are as follows:

(1) We introduce **BeliefConf**, a novel metric to quantify model's confidence towards specific knowledge path.

(2) We design the method of **APAM** (Automated Path Annotation Mechanism) based on comprehensive therotical analysis. Both our preliminary experiments and final experimental results validate the effectiveness of this mechanism, demonstrating its capability to enable LLMs' self-evolution without relying on external supervision.

(3) We propose an efficient framework for ontology-based self-evolution of LLMs in specialized domain, validated through our experiments in the medical domain.

2 Preliminary

2.1 Ontology

Ontology is a type of structured framework that captures concepts, their interconnections, and rules within a specific domain which enables a shared understanding of a domain's knowledge. It has been widely applied in the semantic web and knowledge management systems. Three core components in ontologies are: (1) Concepts: representing entities or categories within a domain. For example, in a medical ontology, concepts might include "Cell", "Symptom" and "Treatment." (2) Relationships: Relationships define how concepts are interconnected. The most common and important relationships in ontologies are: Hyponymy (Is-subclass-of): This represents a hierarchical, subclass relationship. For instance, "Muscle Cell" is a subclass of "Cell" ." Synonymy (Is-synonym-of): This indicates that two concepts are semantically equivalent. For example, "Muscle Cell" and "Muscle Fiber" are synonyms. (3) Axioms (Rules): Ontologies are equipped with built-in rules which enable automated reasoning and consistency checking within knowledge graphs. For example: If (Concept A, Is-subclass-of, Concept B) and (Concept B Is-subclass-of Concept C), then it logically follows that (Concept A, Is-subclass-of, Concept C). However, if the ontology also includes (Concept A Is-Not-A-subclass-of, Concept C), this creates a conflict with the previously inferred relationship. Such rules allow ontologies to automatically detect and resolve inconsistencies, ensuring the integrity of the knowledge graph. This capability is particularly valuable in large-scale knowledge bases, where manual verification would be impractical.

2.2 Perplexity

Model perplexity plays a critical role in our calculation of **BeliefConf**. This metric quantifies the uncertainty of a probabilistic model in its predictions, where lower perplexity values correspond to higher prediction accuracy, while higher values indicate poorer performance. Formally, perplexity is defined as the exponential of the cross-entropy between the true distribution:

$$Perplexity = 2^{H(p,q)} \quad (1)$$

where $H(p, q)$ is the cross-entropy between the true distribution between the true distribution p and the model's predicted distribution q.

The average perplexity of a model on a dataset serves as a proxy for evaluating how well the model comprehends the underlying patterns in the data. Furthermore, next-token prediction perplexity has been adopted to gauge a model's familiarity with specific knowledge items (e.g., Moskvoretskii et al., 2024, Li et al., 2024). In our framework, we leverage this next-token prediction perplexity to compute **BeliefConf**, enabling targeted identification of knowledge gaps for model refinement.

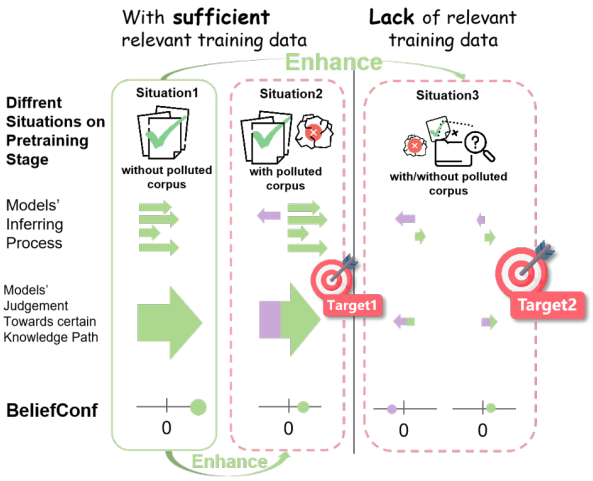

Figure 1: Theoretical analysis of our methodology.

3 Methodology

3.1 Theoretical Analysis and Overview of our Methodology

3.1.1 Theoretical Analysis of APAM

Without access to external annotated knowledge, it is essential to fully leverage the domain-specific knowledge embedded within the model, which is learned from the pretraining stage. To utilize different kind of implicit knowledge critically, we categorize three distinct pretraining scenarios, as illustrated in Figure 1, drawing on insights from Xu et al. (2024). As depicted in the figure, while the majority of the corpus used during pretraining is consistent and accurate, there exists a subset of noisy or polluting data that can undermine the model's confidence in certain knowledge. Additionally, when the pretraining corpus lacks sufficient coverage of certain domain knowledge, the model's confidence in such knowledge tends to be low.

However, we can infer that if multiple detected paths (represented by the green arrows in the figure) consistently support a particular direction, the likelihood of that direction being correct increases, even in the presence of a few conflicting or incorrect paths. This observation forms the basis of our proposed method, **APAM (Automated Path Annotation Mechanism)**.

3.1.2 Overview of our pipeline

Our approach involves the following steps: First, we quantify the model's internal confidence in each knowledge path. Second, we identify whether there are sufficient knowledge paths that support the same conclusion, labeling such paths as reliable paths. Next, we infer new paths based on these reliable paths and evaluate whether the model is already familiar with them. If the model lacks familiarity with these inferred paths, we hypothesize that the model may be encountering either Situation 2 or Situation 3 (as defined in Figure 1), which disrupts its ability to accurately judge the reliability of these paths. We classify such paths—those inferred from reliable paths but unfamiliar to the model—as enhanced paths. Finally, we generate targeted training corpora based on these enhanced paths to refine and improve the model's performance.

3.2 Step 1: Extracting Domain Ontology Framework from original Model

In our methods, we first extract domain ontology from original model, which reflects the model's original belief towards domain concepts and their relationships.

To initiate the generation of the ontology, we manually select seven root concepts in the medical field from the Unified Medical Language System (UMLS)[1]. These root concepts are "Antibiotic", "Bacterium", "Cell", "Hormone", "Tissue", "Vertebrate", and "Vitamin", which are chosen based on their suitability in terms of hierarchical depth and the desired scale of the ontology nodes.

We meticulously design a prompt template as shown in Figure 2 that instructs the model to iteratively generate subclasses and their corresponding synonyms, starting from the concepts at the previous layer. Specifically, the zeroth-layer concepts are the root nodes manually selected in the previous step, while the first- and second-layer concepts are automatically generated by the model based on these roots. These generated concepts then serve as parent nodes for the second- and third-layer concepts, respectively.

To facilitate subsequent processing, we classify the generated nodes according to their layer and generation source(i.e., whether they are derived as subclasses or synonyms), then organize them into a four-level Ontology tree. The edges within the tree are classified into two types: "subclass" edges and "synonym" edges. For a detailed illustration of the node types and the overall structure of the ontology tree, please refer to Figure 3.

[1] https://www.nlm.nih.gov/research/umls/index.html

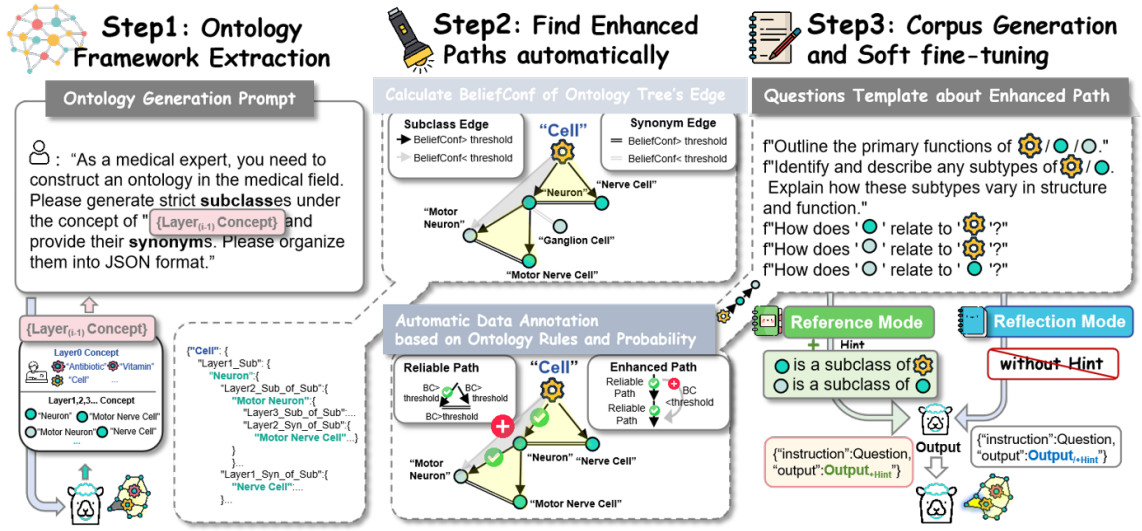

Figure 2: The framework of our method.

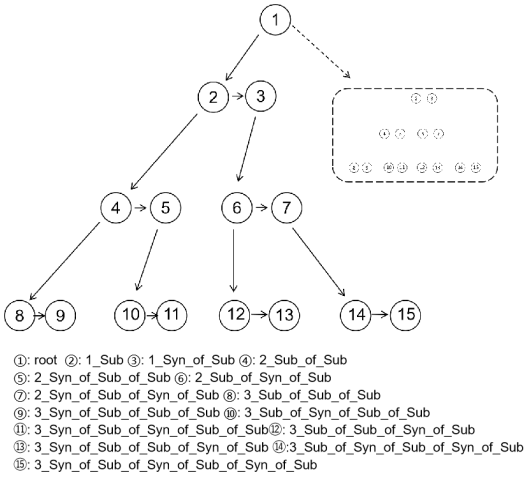

Figure 3: Abstract form of the ontology tree and node classification.

3.3 Step 2: APAM (Automated Path Annotation Mechanism)

3.3.1 Calculation of BeliefConf

Through the generation of the ontology, we can only grip a rough understanding of model's internal domain knowledge system (which may also suffer from hallucination issues in the one-off generation). However, as illustrated in Section 3.1, the model's level of certainty regarding specific knowledge is critically important. To support the following process of **APAM**, it is essential for us to obtain a quantitative measure of this certainty. Thus, we introduce the metric called **BeliefConf**, and we calculate BeliefConf of every edge in the generated ontology tree.

Preparation for Calculating BeliefConf: Computing the Perplexity of Each Path Perplexity is a widely used metric serving as an indicator of the model's certainty in predicting the next token. We leverage this metric to evaluate the model's confidence towards every knowledge paths in the Ontology Tree constructed in the previous step.

To mitigate the potential influence of different hypernym-hyponym concepts within a sentence on the overall perplexity, we adopt a next-token prediction approach for perplexity calculation. Specifically, we design two types of prompts: one representing "Support" and the other "Against" . These prompts are identical in structure, differing only in the final token of the Answer section—"True" for the "Support" prompt and "False" for the "Against" prompt.

Intuitively, by comparing the perplexity of the final token in the "Support" and "Against" prompts, we can infer the model's belief towards a given piece of knowledge.

Precise Definition of BeliefConf To compare the model's confidence levels across different pieces of knowledge, we assume that when the smaller perplexity between the "True" and "False" options is even smaller, or the larger perplexity is even larger, or when the gap between the "True" and "False" perplexities(ppl) is wider, it indicates that the model has a better understanding of the relationship and greater confidence in judging the correctness of the knowledge.

Based on this intuition, we calculate the minimum, maximum, and difference values of the

true_ppl and false_ppl, and propose the following three definitions of JC(Judge Confidence):

$$JC_{min} = \min(true\_ppl, false\_ppl) \quad (2)$$

$$JC_{max} = \max(true\_ppl, false\_ppl) \quad (3)$$

$$JC_{diff} = |(true\_ppl - false\_ppl)| \quad (4)$$

The definition of Judge Confidence solely reflects the model's confidence in judging a particular piece of knowledge. To determine the model's final qualitative judgment—whether the knowledge is "true" or "false"—we further compare the difference between true_ppl and false_ppl. In the selection of reliable paths, only those short paths where the Judge Confidence exceeds a predefined threshold and the model's final qualitative judgment is "true" can form "strongly supportive" edges, which are eligible to connect into a coherent path. Conversely, "strongly opposed" edges, which exhibit high Judge Confidence but are ultimately judged as "false," cannot be included in the construction of long paths.

Building on this, we introduce a precise definition of BeliefConf, which quantifies the model's degree of support for a given piece of knowledge. As detailed in Section 3.1, when both the BeliefConf of two short paths and the BeliefConf of the long path they form all exceed the threshold, we designate the long path as a Reliable Path. However, if the BeliefConf of the long path formed by two short paths (marked as Reliable Paths) falls below the threshold, we infer that the model lacks sufficient familiarity with the concepts involved in the path. In such cases, we label the long path as an Enhanced Path.

It is worth noting that to further validate the rationality of the BeliefConf calculation, we sampled 700 paths each for three calculation methods: min, max, and minus, and used GPT-4o-mini to evaluate the model's judgment accuracy. The results show a positive correlation between BeliefConf and the model's accuracy.

$$BC_{min} = \begin{cases} \frac{1}{JC_{min}}, if support > 0; \\ -\frac{1}{JC_{min}}, if support < 0. \end{cases} \quad (5)$$

$$BC_{max} = \begin{cases} JC_{max}, if support > 0; \\ -JC_{max}, if support < 0. \end{cases} \quad (6)$$

$$BC_{minus} = \begin{cases} JC_{minus}, if support > 0; \\ -JC_{minus}, if support < 0. \end{cases} \quad (7)$$

3.3.2 Threshold Setting

After defining the evaluation metric **BeliefConf** to assess the model's endorsement of a path, it is necessary to establish a threshold. This threshold allows us to label long paths as Reliable Paths when the BeliefConf of its constituent short paths and the long path itself exceeds the threshold. Additionally, for knowledge edges that are logically inferred by reliable paths but fall below the threshold, we mark them enhanced paths and make targeted improvements.

Based on intuition and preliminary experiments, we observe that stricter threshold settings lead to higher factual accuracy of the filtered knowledge but simultaneously reduce the number of training knowledge retained. In this study, we balance the trade-off between the sufficiency of training instances and the accuracy of the knowledge by considering six threshold calculation methods: the top 10%, 20%, 30%, 40%, and 50% quantile values, as well as the mean value. We show the trade-off details in the appendix.

3.3.3 Filtering Modes for Reliable Paths and Enhanced Paths

As described in previous sections, we have obtained a four-level Ontology structure containing multiple concepts through Step 1, as illustrated in Figure 3. Additionally, through Step 2, we have calculated the BeliefConf for each edge in the structure. In the following, we will apply threshold filtering to identify Reliable Paths and Enhanced Paths within this structure.

Reliable Path Filtering In order to identify sufficient knowledge paths for training, we aim to obtain one-hop hyponym-hypernym relationships as Reliable Path. To achieve this, we leverage synonyms to serve as the second short-path edge.

As illustrated in Figure 4, for the hypernym node 1, the model generates the hyponym node 2, along with several synonym nodes of node 2 (i.e., nodes 2.1, 2.2, and 2.3). We first identify hyponym edges (green arrow in the left part of the figure, connecting node 1 and node 2 where the BeliefConf of the hypernym-hyponym relationship exceeds the threshold. Next, among all co-hyponym relationships of node 2, we locate synonym edges (double green lines in the right part, connecting node 2 and node 2.1) where the BeliefConf also exceeds the threshold. Finally, we manually add an edge connecting node 1 and node 2.1, referred to as a

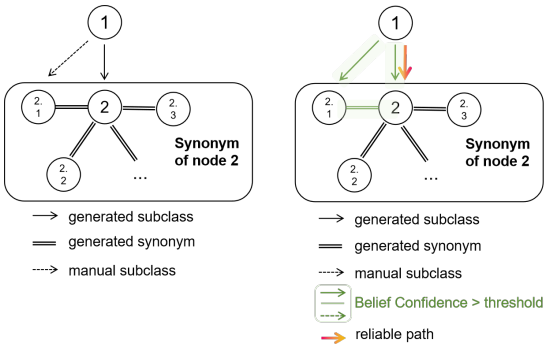

Figure 4: Selection of reliable path.

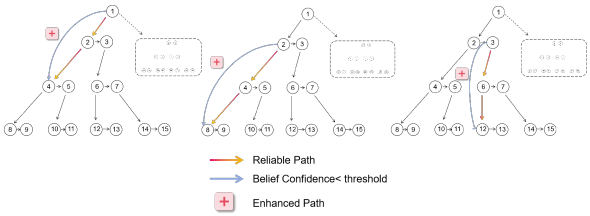

Figure 5: Selection of enhanced path.

"manual subclass" edge.

It is important to note that the addition of the "manual subclass" edge is based on ontology rule: (Concept B, is a subclass of, Concept A), (Concept C, is a synonym of, Concept B) → (Concept C, is a subclass of, Concept A). Although this edge is not directly generated by the model, we can infer the implicit hypernym-hyponym relationship between these concepts using Ontology rules. We then calculate the BeliefConf for this "manual subclass" edge using the aforementioned method.

If the BeliefConf of this "manual subclass" edge exceeds the threshold, which means that the BeliefConf of all edges connecting nodes 1, 2, and 2.1 is above the threshold, we label the one-hop hypernym-hyponym relationship (1 → 2) as a Reliable Path.

Enhanced Path Filtering After labeling several one-hop hypernym-hyponym relationships as Reliable Paths, we identify the following two-hop long paths formed by the chaining of two Reliable Paths: 1 → 4, 2 → 8, and 3 → 12 (See in Figure 5). We then evaluate whether the BeliefConf of these long paths falls below the predefined threshold. If it does, we label them as Enhanced Paths that require supplementation.

3.4 Step3: Fine-tuning Corpus Generation

3.4.1 Fine-tuning Settings

Currently, we have identified the edges that require enhancement, referred to as Enhanced Paths. We hypothesize that the model is less familiar with the concepts and hypernym-hyponym relationships involved in these edges. Based on previous work(Zhang et al., 2024), we recognize the importance of the naturalness and richness of training corpora. Therefore, we have designed five contextualized template prompts(See in Figure 2) for corpus generation, into which the concepts associated with the Enhanced Paths are inserted. These template prompts not only address the relationships between concepts but also explore the characteristics of the concepts themselves, such as their structure or function, to simultaneously improve the model's understanding of both the relationships between concepts and the concepts themselves. Additionally, previous research(Tao et al., 2024) has demonstrated that language models can efficiently self-evolve through self-generated corpus. Thus, we allow the model to generate answers to these prompts itself, and these self-generated responses are then used for the model's self-training.

Our question templates are exhibited in Figure 2:

We designed two fine-tuning scenarios: Reflection Mode without ontology hint and Reference Mode with ontology as hint in the prompt for corpus generation.

3.4.2 Reflection Mode without ontology hint

In this scenario, the model is directly provided with the aforementioned question templates as input and is asked to generate responses by using its existing knowledge. This process requires the model to reorganize and reflect on these unfamiliar concepts independently.

3.4.3 Reference Mode with ontology as hint

This setup appends a Hint containing the Enhanced Path to the question template. The goal is to assist the model in reflecting on these concepts by providing references. To avoid potential negative impacts from incorrect paths, we implement a friendly reminder, "You can consider these relationships as follows, but please ignore them if they are unnecessary." before the ontology hint.

The fine-tuning corpus format for each scenario is showed in Figure 2.

Model	Medical						
	PubMedQA	MedQA	MedMCQA	USMLE- step1	USMLE- step2	USMLE- step2	Average
Llama3-8b-instruct	67.4	49.3	49.2	56.4	50.5	60.7	55.6
Llama3-Aloe-8B-Alpha	65.8	35.6	40.5	39.4	40.4	45.1	44.5
Llama3-Med42-8B	66.5	**56.2**	**56.9**	**61.7**	**60.1**	**65.6**	**61.2**
jsl-MedLlama-3-8B-v2.0	59.5	24.4	42.6	24.5	22.0	23.0	32.7
ours	**69.8**	52.1	48.3	57.4	50.5	59.8	56.4

Table 1: Main results.

4 Experiments

4.1 Experimental Setup

Given its broad applicability and significance, we select the medical domain for experiments.

Dataset and Metrics. Following previous experiments(Liu et al., 2025; Christophe et al., 2024), we select several representative medical-domain datasets and comprehensively evaluate the model's performance across various medical tasks.. These include: PubMedQA, MedQA, MedMCQA(Pal et al., 2022) and USMLE step1-3 datasets. In the ablation study, we also use lmharness[2] to evaluate model's ability on PubMedQA and MMLU.

Baselines. We compare our approach against several models fine-tuned from the same base model, LLaMA3-8B-Instruct, including Aloe(Gururajan et al., 2024), Med42-v2-8B, and jsl-MedLlama-3-8B-v2.0[3]. Additionally, we include the baseline LLaMA3-8B-Instruct model for comparison. In the ablation study, we also use Taxollama(Moskvoretskii et al., 2024) as a baseline, which injects ontology paths directly into the model without generating additional training corpora.

Implementation and Variants of our model. We use LLaMA3-8B-Instruct as the foundation model for self-evolution. Fine-tuning is conducted using the Llamafactory(Zheng et al., 2024) framework, with the LoRA (Low-Rank Adaptation) method for parameter-efficient training. All experiments are performed on NVIDIA A800 80GB GPUs, with a learning rate of 5e-5, trained for 3 epochs using a cosine scheduler.

By adopting different BeliefConf and threshold settings, the number of ReliablePath and EnhancePath instances, as well as the their estimated accuracy varies, which impacts the model's performance. In Table 1, we report the results of the best-performing model variant. This model uses a threshold setting of 50th percentile and is trained on a total of 11000 data instances. In the ablation study, we explore other threshold settings and their effects on performance. Beyond EnhancePath, we further investigate the impact of training with paths that the model is already familiar with (i.e., paths with BeliefConf above the threshold).

[2] https://github.com/EleutherAI/lm-evaluation-harness

[3] https://https://huggingface.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0

pathtype	traintype	pubmedqa	mmlu
Llama3-8b-instruct		74.6	63.84
Enhanced Path	withonto	76.0	63.40
	withoutonto	75.2	63.67
	taxollama	74.6	63.76
Convinced Path	withonto	74.6	63.69
	withoutonto	74.6	63.82
	taxollama	74.2	63.57

Table 2: Ablation study on different path types and training corpus.

4.2 Main Result

As illustrated in Table 1, our model achieves an average score of 56.4, ranking second overall among the compared models. It demonstrates competitive performance across multiple tasks, particularly excelling in PubMedQA (69.8) and USMLE-step1 (57.4). However, there is room for improvement in tasks like MedMCQA (48.3), where it falls slightly behind the top-performing model. Although our model lags behind Med42, which is fine-tuned on a large corpus, on most datasets, it outperforms the base model on 3 out of 6 datasets and achieves comparable performance to the base model on the USMLE-Step2 dataset. Notably, our model surpasses Med42 on the PubMedQA, achieving the best performance without relying on external data. This confirms the effectiveness of our approach.

4.3 Further Analysis

Model's Familiarity with the Supplemented Paths To investigate whether addressing the unfamiliarity of Enhanced Paths improves the model's performance, we define Convinced Paths as long paths where both the long path and its two con-

stituent short paths have BeliefConf values exceeding the threshold. We set a threshold such that the number of Enhanced Paths and Convinced Paths is equal. The results in Table 2 demonstrate that models trained with Enhanced Paths exhibit significantly better performance than those trained with Convinced Paths. This highlights the effectiveness of point-to-point knowledge correction in addressing the model's knowledge gaps.

Different Modes of Corpus Generation We experimented with different modes of corpus generation, as described earlier. The results for one threshold combination are presented in Table 2. They show that when Enhanced Paths are used, models fine-tuned with reference mode perform significantly better than those fine-tuned with reflection mode. In contrast, for Convinced Paths, there is no significant difference between the two modes. This aligns with our intuition: Enhanced Paths represent knowledge the model is unfamiliar with, so providing additional context (via ontology hints) is beneficial, whereas Convinced Paths represent knowledge the model is already confident about, making additional context less impactful.

However, when evaluating the models on the MMLU dataset, we observe that all fine-tuned models exhibit a decline in performance. Notably, models trained reflection-mode corpus show a smaller decline, suggesting that injecting unfamiliar information via with-ontology prompts may slightly hinder the model's general capability.

Ablation Study of Corpus Generation We also compared our method with TaxoLLaMA in Table 2. The performance gap using Taxollama between Enhanced Paths and Convinced Paths aligns with our findings. However, models trained with TaxoLLaMA perform weaker than our models even though the number of paths was controlled to be the same, likely due to differences in the quantity and quality of the generated corpora.

5 Related Works

Domain Fine-Tuning Models. Domain Fine-Tuning Models refer to the process of further training pre-trained models on domain-specific data to adapt them to the tasks and linguistic characteristics of the target domain. This approach combines the general knowledge of pre-trained models with domain-specific expertise, significantly enhancing their performance on specialized tasks. While different training strategies—such as Full Fine-Tuning or Low-Rank Adaptation —may be employed, most models rely on large-scale domain-specific datasets for optimization(Gururajan et al., 2024; Christophe et al., 2024). For example, Aloe uses 348K medical QA pairs from 20+ sources, 430K synthetic medical QA pairs and 122K high-quality general-domain samples for fine-tuning.

Self-evolution of LLM. Model Self-Evolution refers to the process of leveraging large language models (LLMs) to autonomously generate high-quality training data without relying on external annotations, effectively mitigating domain data sparsity issues while reducing resource costs such as human effort. To achieve robust self-evolution, two critical challenges must be addressed: Firstly, How to automatically identify knowledge requiring self-improvement. Secondly, How to determine which generated data is of higher quality in the absence of external labels, and enable the model to learn from it. For the first challenge: Wang et al., 2023b guides models to generate their own task instructions and corresponding responses, thereby enhancing their ability to handle such instructions. For the second challenge, many approaches involve defining human-crafted principles to guide the model in selecting higher-quality knowledge. For example, Huang et al., 2023,Wang et al., 2023a,Madaan et al., 2023)prioritize consistency. Sun et al., 2023 guides models to generate outputs adhering to predefined criteria such as ethics or informativeness , and Yu et al., 2024 employs chain-of-thought (CoT) reasoning to introduce higher quality answers.

6 Conclusion

We propose a framework for ontology-based self-evolution of LLMs, leveraging the Automated Path Annotation Mechanism (APAM) and the BeliefConf metric to enhance domain-specific knowledge without external supervision. Experiments in the medical domain show our method outperforms the base model (Llama3-8B-instruct) on 3 out of 6 datasets (PubMedQA, MedQA, and USMLE-step1) and achieves state-of-the-art performance on PubMedQA. This work bridges LLMs with symbolic reasoning-based knowledge graphs, enabling models's self-evolution in specialized domains. In the future, we aim to introduce more robust path annotation patterns into this framework and we hope that this framework can be adapted to more domains suited for ontology-based approaches.

Limitations

Limitations in Feature Recognition and Completion Patterns Due to time constraints, we have only focused on completing two-hop hyponymy relations. In practice, the same approach could be applied to obtain reliable paths for synonyms, thereby enhancing one-hop hypernymy-hyponymy relations, among others. Additionally, the model's understanding of a particular piece of knowledge may depend on more features, such as the familiarity of neighboring edges or the conceptual proximity of related terms. This paper, however, only considers a single feature: the perplexity of one-hop paths. We have conducted only preliminary experiments on feature recognition and completion patterns to explore the feasibility of a framework for model self-evolution using Ontology. More comprehensive experiments will be reserved for future work.

Potential Error Risks Since this method aims to achieve model self-evolution without external supervision, there is an inevitable risk of introducing incorrect knowledge when applying fine-tuning settings with referenced model reflection. Although we have implemented a Reliable Path safeguard system and a fine-tuning corpus generation method with cautionary prompts to minimize potential errors, there is no guarantee of the complete correctness of unsupervised data. In future research, we will conduct a more detailed analysis of how risky or erroneous Paths might affect model performance and explore ways to enhance the model's ability to mitigate such risks.

Broader Applicability Currently, we have only explored the applicability of this method in medical domain. And we have utilized only two common types of relations in ontology: "is-subclass-of" and "is-synonym-of." In more specialized or even general domains, there exist similar or more diverse explicit relational rules. The application of the proposed approach in these domains will be left for future research.

References

Clément Christophe, Praveen K. Kanithi, Tathagata Raha, Shadab Khan, and Marco AF Pimentel. 2024. Med42-v2: A suite of clinical llms. *CoRR*, abs/2408.06142.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Ashwin Kumar Gururajan, Enrique Lopez-Cuena, Jordi Bayarri-Planas, Adrián Tormos, Daniel Hinjos, Pablo Bernabeu-Perez, Anna Arias-Duart, Pablo Agustin Martin-Torres, Lucia Urcelay-Ganzabal, Marta Gonzalez-Mallo, Sergio Álvarez-Napagao, Eduard Ayguadé Parra, and Ulises Cortés Dario Garcia-Gasulla. 2024. Aloe: A family of fine-tuned open healthcare llms. *CoRR*, abs/2405.01886.

Tianyu Han, Lisa C. Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K. Bressem. 2023. Medalpaca - an open-source collection of medical conversational AI models and training data. *CoRR*, abs/2304.08247.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *EMNLP*, pages 1051–1068. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP/IJCNLP (1)*, pages 2567–2577. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *NAACL-HLT*, pages 7602–7635. Association for Computational Linguistics.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun-Qing Li, Hejie Cui, Xuchao Zhang, Tian yu Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, Carl Yang, and Liang Zhao. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey.

Zhiqiang Liu, Chengtao Gan, Junjie Wang, Yichi Zhang, Zhongpu Bo, Mengshu Sun, Huajun Chen, and Wen Zhang. 2025. Ontotune: Ontology-driven self-training for aligning large language models. *arXiv preprint arXiv:2502.05478*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*.

Viktor Moskvoretskii, Ekaterina Neminova, Alina Lobanova, Alexander Panchenko, and Irina Nikishina. 2024. Taxollama: Wordnet-based model for solving multiple lexical sematic tasks. *CoRR*, abs/2403.09207.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *CoRR*, abs/2203.14371.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *NeurIPS*.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. 2024. A survey on self-evolution of large language models. *CoRR*, abs/2404.14387.

Mengru Wang, Yunzhi Yao, Ziwen Xu, Shuofei Qiao, Shumin Deng, Peng Wang, Xiang Chen, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. Knowledge mechanisms in large language models: A survey and perspective. In *EMNLP (Findings)*, pages 7097–7135. Association for Computational Linguistics.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *ACL (1)*, pages 13484–13508. Association for Computational Linguistics.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey. In *EMNLP*, pages 8541–8565. Association for Computational Linguistics.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *CoRR*, abs/2407.06023.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *CoRR*, abs/2403.13372.

A The distribution of BeliefConf

The distribution of BeliefConf We evaluated the distributions of BeliefConf (max), BeliefConf (min), and BeliefConf (minus) across over thirty thousand hyponymy and synonymy relations. The illustrative distributions are presented in Figures 6, 7, and 8.

B Hyperparameter Analysis for Reliable Edge Threshold Filtering.

In Table 3, we present the evaluation results of reliable path quantities and GPT-4o-mini's accuracy

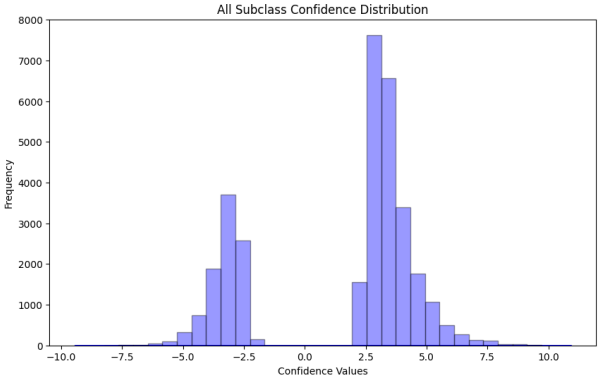

Figure 6: Distribution of Belief$_{max}$ in subordinate relationships.

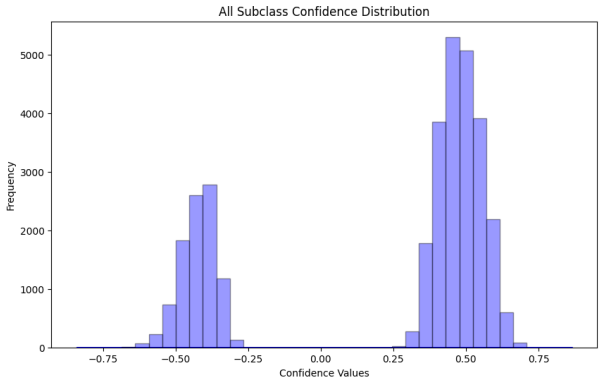

Figure 7: Distribution of Belief$_{min}$ in subordinate relationships.

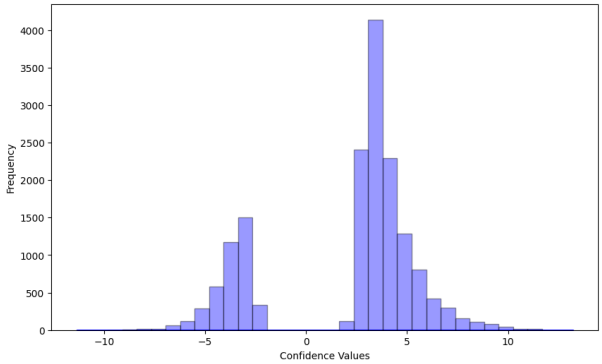

Figure 8: Distribution of Belief$_{diff}$ in synonym relationships.

rates corresponding to several threshold filtering combinations.

Threshold	Num of Reliable Path	Estimated Accuracy
combo1	1909	82.3%
combo2	3759	75.3%
combo3	3424	74%
combo4	6903	66%

Table 3: Hyperparameter analysis on threshold.