From Experts to LLMs: Evaluating the Quality of Automatically Generated Ontologies

Majlinda Llugiqi[1,*], Fajar J. Ekaputra[1] and Marta Sabou[1]

[1]Vienna University of Economics and Business, Vienna, Austria

Abstract

Ontologies play a crucial role in knowledge representation, yet their manual construction requires domain expertise and effort. While previous work has focused on using large language models (LLMs) for assessing ontology creation, fully automated ontology generation remains underexplored. As a consequence, most research relies on a limited set of well-known ontologies or knowledge graphs, which constrains the evaluation of various tasks such as link prediction and knowledge graph completion. This highlights the need for diverse ontology benchmarks with varying characteristics, such as number of concepts, hierarchy depth and so on, to effectively evaluate tasks such as link prediction and knowledge graph completion. In this work, we investigate the feasibility of generating ontologies using LLMs and evaluate whether they can produce ontologies of comparable quality to human-built ones. Given a seed set of concepts, a target number of concepts, relations, and maximum hierarchy depth, we employ three different LLMs to generate ontologies within the heart disease domain. Defining a seed set of concepts is particularly important for modeling the features of tabular datasets, enabling structured knowledge representation for downstream tasks. We systematically evaluate the generated ontologies by analyzing their structural integrity, semantic coherence, and suitability for downstream tasks. Our results show that while LLM-generated ontologies differ structurally from human-built ones, they remain comparable in semantic similarity and downstream ML performance, with LLaMA-generated ontologies proving to be the most effective. These findings highlight the potential of LLM-generated ontologies not only to support automated knowledge representation but also to enhance ontology benchmarks by introducing diverse structural characteristics, enabling more comprehensive evaluations of machine learning tasks.

Keywords

Ontology Generation, Domain-Specific Ontologies, Large-Language Models, Ontology Evaluation

1. Introduction

Ontologies play a crucial role in structuring knowledge across various domains, enabling tasks such as semantic reasoning, data integration, and knowledge representation. However, ontology construction methods rely heavily on expert knowledge and manual effort, which can be time-consuming, costly and difficult to scale [1]. Moreover, there is a lack of automated methods for generating domain-specific ontologies that accurately represent datasets' features [2]. Most research and applications depend on well-known ontologies and knowledge graphs, which, while useful, do not always offer the flexibility needed for evaluating diverse tasks [3]. In particular, there is a need for ontologies with varying structural characteristics to enable a more comprehensive evaluation of Machine Learning (ML) tasks such as link prediction and knowledge graph completion [4].

Several efforts have explored the use of large language models (LLMs) to assist ontology engineering [1, 5, 6, 7], aiming to address the gap mentioned above. These include generating competency questions, completing missing ontology components, and supporting ontology alignment. However, existing approaches do not provide a fully automated pipeline for generating high-quality ontologies from scratch, containing different characteristics. To address the need for ontologies with different characteristics, recent work has introduced tools such as PyGraft [3], which can generate synthetic ontologies with predefined structural properties such as the number of their concepts and relations. Although these

2nd International Workshop on Evaluation of Language Models in Knowledge Engineering Co-located with the Extended Semantic Web Conference (ESWC 2025)

*Corresponding author.

majlinda.llugiqi@wu.ac.at (M. Llugiqi); fajar.ekaputra@wu.ac.at (F. J. Ekaputra); marta.sabou@wu.ac.at (M. Sabou)

0000-0002-5008-6856 (M. Llugiqi); 0000-0003-4569-2496 (F. J. Ekaputra); 0000-0001-9301-8418 (M. Sabou)

tools are valuable for benchmarking and experimentation, they produce domain-agnostic ontologies, making them less suitable for practical applications where domain knowledge is essential.

Addressing these challenges, our work investigates the potential of LLMs to generate domain-specific ontologies. Given a set of seed concepts, along with constraints on the number of concepts, relations, and maximum depth, our approach aims to construct structured ontologies that accurately capture domain knowledge, modeling datasets' features while allowing flexibility in their characteristics. Our research is guided by the following research questions:

- RQ1: To what extent do LLM-generated ontologies align with human-built ontologies in terms of domain relevance, hierarchical organization and performance on downstream tasks?
- RQ2: How to evaluate LLM generated ontologies?
- RQ3: Which LLM performs best in generating ontologies that closely resemble human-built ones?

To address these research questions, we used a methodology that systematically compares LLM-generated ontologies with a human-built ontology in heart disease domain. Specifically, we focused on an ontology designed to represent heart-disease dataset's features [1], which contain clinical and diagnostic features relevant to cardiovascular conditions (e.g., chest pain, heart rate). We generated three ontologies using three different LLMs, each prompted with the same request to produce an ontology incorporating predefined medical terms, a specified number of concepts and relations, and a maximum hierarchical depth. These were evaluated against the human-built ontology using structural (e.g., average degree, path length, branching factor), semantic (e.g., information content, Jaccard similarity for classes and relations, ontology embeddings) comparisons, as well as task-based evaluation.

In the task-based evaluation, we followed [8, 9] approach, transforming the ontologies into knowledge graphs by populating them with dataset instances and generating knowledge graph embeddings using four different embedding methods. We then computed two embedding-based metrics. These metrics were used to augment the original tabular dataset, and then evaluated four ML models for binary classification. While in these previous works exclusively human-built ontologies were used, our study extends this evaluation to LLM-generated ontologies to assess whether they can support downstream tasks as effectively as human-built ones. Performance was measured using accuracy and F2-score, providing insights into their applicability in machine learning contexts.

Our main contribution is a new method for ontology engineering that allows for the generation of ontologies with diverse structural and semantic characteristics, along with a systematic evaluation framework for assessing LLM-generated ontologies. Our framework evaluates these ontologies through structural, semantic, and task-based comparisons. We analyze graph-based properties to evaluate hierarchical organization, use Jaccard similarity, information content, and ontology embeddings for semantic alignment, and assess real-world applicability by evaluating their impact on a binary classification task. Results show that while LLM-generated ontologies exhibit structural differences from human-built ones, they remain comparable in semantic similarity and task-based performance. Among them, LLaMA-generated ontologies prove to be the most effective, in some cases even outperforming human-built ontologies in downstream tasks. Additionally, both human-built and LLaMA-generated ontologies tend to form more structured knowledge hierarchies, whereas GPT and DeepSeek-generated ontologies show different structural biases, GPT favoring flatter structures and DeepSeek generating deeper, more linear taxonomies. Furthermore, ontology embeddings from LLaMA closely align with those of human-built ontologies, whereas GPT and DeepSeek-generated embeddings show greater discrepancies, suggesting a weaker preservation of conceptual relationships.

The rest of the paper is organized as follows. In Section 2, we discuss the related work, followed by Section 3 where we present the methodology that we use for ontology generation and evaluation. Further in Section 4 we discuss the experiment setup, continued by the results in Section 5. Finally, we summarize our findings and outline directions for future work in Section 6.

[1] https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

2. Related Work

In this section we review related work on ontology evaluations methods and LLMs for ontology generation.

Ontology Evaluation Methods Ontology evaluation has been widely studied in knowledge representation, with various methodologies developed to assess their quality, consistency, and applicability. Traditional methods include crowdsourcing approaches [10] and expert evaluation, both of which offer high-quality assessments but can become costly and impractical when evaluating large and multiple ontologies. To mitigate this, automated reasoning techniques such as HermiT [11] and OntoClean [12] have been employed to ensure logical consistency and conceptual clarity in ontology development. These approaches allow for automated verification of taxonomic and logical constraints, reducing the reliance on human intervention. Additional evaluation metrics include completeness, graph-based structural properties, and domain coherence.

More recently, LLMs have been leveraged for ontology evaluation. Tsaneva et al. [13] proposed an LLM-driven approach for verifying ontology restrictions, demonstrating that LLMs can achieve intermediate-to-expert performance levels on ontology modeling qualification tests.

In addition to assessing structural correctness, ontology evaluation also considers the suitability of semantic resources, such as ontologies and knowledge graph (KG) for downstream applications, such as knowledge graph embeddings (KGEs) [14]. This suggests that ontology evaluation should extend beyond correctness checks and incorporate empirical validation through task-based performance assessments.

Raad et al. [15] categorized ontology evaluation methods into four groups: gold standard-based, corpus-based, task-based, and criteria-based approaches. The integration of LLMs into these methodologies represents a promising direction for improving ontology validation, particularly in cases where conventional expert-driven evaluation is infeasible.

In this paper, we use three different ways to evaluate the ontologies generated with LLMs. Given our focus on the structure of ontologies generated by LLMs under predefined structural constraints, we utilize graph-based structural evaluation metrics, including average degree, path length, branching factor, and degree distribution. Furthermore, since we aim to generate ontologies containing specific terms for modeling datasets and possess a gold-standard ontology, we incorporate semantic metrics to compare human-generated and LLM-generated ontologies. These metrics include Jaccard similarity, embedding similarity, and information content. Finally, we conduct a task-based evaluation of the ontologies, drawing on recent studies that explore the use of ontologies and KG information to enhance ML prediction [8, 9].

LLMs for supporting knowledge engineering LLMs are increasingly used to support knowledge engineering tasks. One application is the generation of Competency Questions, as demonstrated by [16, 17]. Another growing research area is the automatic generation of ontologies from textual or structured data using LLMs [1, 18, 19, 20]. For instance, [21] evaluate multiple LLMs and prompting strategies for generating OWL ontologies directly from ontological requirements, identifying GPT-4 as the most effective among the models tested. Additionally, LLMs have been explored for knowledge graph completion, where they assist in inferring missing links and enriching structured knowledge bases [22, 23].

Beyond these applications, LLMs have also been leveraged for ontology alignment, where they facilitate the matching of concepts across different ontologies [24], and for ontology population, where they help instantiate knowledge bases with factual data extracted from unstructured text [25]. In the realm of visual activity understanding, hybrid approaches that integrate LLMs with symbolic reasoning have been developed to enhance explainability, generalization, and data efficiency. For instance, the Symbol-LLM framework leverages LLMs to generate broad-coverage symbols and rational rules, facilitating fuzzy logic-based reasoning over visual inputs. [26].

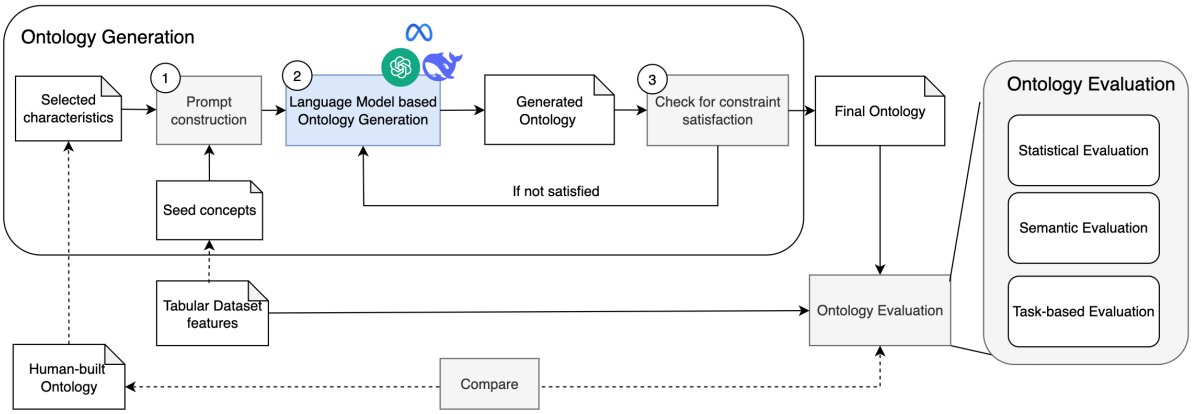

Figure 1: Overview of the methodology used for ontology generation using LLMs and evaluation. The dashed lines indicate optional steps, which are applied when both the dataset and human-built ontologies are available.

Most of the existing work discussed focuses on supporting specific knowledge engineering tasks rather than constructing the entire ontology. In contrast, our work aims to generate ontologies given only a domain specification, a set of seed concepts, and structural constraints. This approach enables the creation of ontologies tailored to represent tabular datasets with defined features. Moreover, it serves as a first step toward systematically generating ontologies with varying structural and semantic properties, which can be used to evaluate different downstream tasks, such as link prediction, knowledge graph completion, and ML tasks.

3. Methodology for LLM-Based Ontology Generation and Evaluation

This section presents our methodology, with Section 3.1 focusing on the generation of ontologies using LLMs, while Section 3.2 describes the evaluation approaches.

3.1. Ontology Generation

As shown in Figure 1, the ontology generation process is organized in three steps: prompt construction, LLM-based ontology generation and checking for constraint satisfaction.

Step 1: Prompt Construction The process begins with the selection of desired characteristics, which define the structure of the ontology, as well as the seed concepts, which are the concepts that we want to model, and might be extracted from tabular dataset features.

For our experiment, we used the Heart Disease ontology from [27] (as discussed in the Section 4) and the Heart Disease Prediction dataset[2]. The characteristics selected for ontology generation included the number of classes (29), relations (6), and max depth (5), which are the characteristics that the human-built ontology has, that may serve as a reference for comparison. For the seed concepts, the heart disease dataset contains 14 features, so we created a corresponding list of 14 terms, using their full descriptive names rather than the abbreviated versions found in the dataset (e.g., as presented on Kaggle), as follows:

[2]https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

terms = [”age”, ”sex”, ”chest pain type”, ”resting blood pressure”, ”serum cholestoral”, ”fasting blood sugar”, ”resting electrocardiographic results”, ”maximum heart rate achieved”, ”exercise induced angina”, ”oldpeak = ST depression induced by exercise relative to rest”, ”the slope of the peak exercise ST segment”, ”number of major vessels colored by fluoroscopy”, ”thallium stress”, ”heartDisease”]

With the selected characteristics and seed concepts defined, we constructed a prompt to guide the LLM-based ontology generation, formatted as follows:

You are a knowledge engineer specializing in ontology design. Your task is to generate a domain-specific ontology for heart disease in Turtle format,with the following properties:

- Use the following terms as seed concepts: `{terms}`
- It must have exactly `{num_classes}` classes, no more, no less.
- It must have `{num_relations}` unique relations, no more, no less, with their domain and range specified.
- The maximum hierarchy depth should be `{max_depth}`.
- If you struggle to fit terms while maintaining `{num_classes}` classes, create additional generic but relevant classes.
- No additional commentary—output, only the ontology.
- Ensure meaningful and logically coherent class relationships.

Think step by step and enumerate the number of classes and relations while structuring the ontology, but in your response, include **only** the final ontology without any explanation, and please make sure that it has the requested amount of classes, relations and max depth.

Step 2: LLM-based Ontology Generation After the prompt construction, we then provide the prompt to three different LLMs (detailed in the Section 4) to generate ontologies.

Step 3: Constraint checking Each generated ontology undergoes an initial validation step to ensure it meets the predefined constraints (e.g., number of classes, relations, and maximum hierarchy depth). If the constraints are not satisfied, the ontology is regenerated iteratively until either the requirements are met or a threshold is reached. Once validated, the final ontology is saved for further evaluation and comparison.

The LLM-generated ontologies used in this study are publicly available at Zenodo [3] to facilitate reproducibility and future research.

3.2. Ontology Evaluation

To assess the quality of the generated ontologies, as illustrated in Figure 1, we employ a three-tier evaluation approach consisting of structural, semantic, and task-based evaluation methods. These evaluations measure structural integrity, conceptual alignment, and practical usability. Finally, the generated ontologies are compared against a human-built ontology.

The metrics used for each evaluation method are detailed in Table 1. Below, we describe each metric in detail.

Structural-Based Methods: Structural-based evaluation focuses on assessing the structural properties of the ontology as a graph. These metrics capture aspects such as connectivity, hierarchy, and distribution of nodes and edges. The structural-based metrics include:

[3] https://zenodo.org/records/15013330

Method	Metric	Explanation
Structural	Avg Degree	Avg. edges per node: $\frac{\sum \text{degrees}}{\text{nodes}}$
	Path Length	Mean shortest path between nodes.
	Branching Factor	Avg. subclasses per class: $\frac{\sum \text{Children}}{\text{Parents}}$
	Degree Dist.	Node degree distribution (connections per class)
Semantic	Jaccard (Classes)	Concept overlap: $\frac{\lvert C_{\text{human}} \cap C_{\text{AI}} \rvert}{\lvert C_{\text{human}} \cup C_{\text{AI}} \rvert}$
	Jaccard (Relations)	Relation overlap: $\frac{\lvert R_{\text{human}} \cap R_{\text{LLM}} \rvert}{\lvert R_{\text{human}} \cup R_{\text{LLM}} \rvert}$
	Ontology Embedding Sim.	Node2Vec-based cosine similarity.
	Information Content	Generality or specificity of concepts. $1 - \frac{\log(\lvert \text{descendants}(C) \rvert + 1)}{\log(\lvert C \rvert)}$
Task	DistAugTab	Tabular dataset enrichment with euclidean distance of embeddings to target class centroids.
	EmbedClustAugTab	Tabular dataset enrichment with embedding vectors and k-means cluster membership.

Table 1
Overview of the ontology metrics used for evaluation.

- Average degree, which measures the average number of edges (relations) per node (concept), providing insights into overall connectivity of the ontology.
- Path length, which measures the mean shortest path between ontology concepts, showing how the concepts are connected. If the ontology is not connected, it computes the average path over weakly connected components.
- Branching factor, which computes the average number of subclasses per class, indicating the hierarchy's complexity.
- Degree distribution, which measures the spread of connections per class, helping understand the potential structural imbalances.

These metrics help compare the structure of generated ontologies against human-built ones, ensuring that LLM-generated ontologies maintain a reasonable level of hierarchy and connectivity.

Semantic-Based Methods: Semantic-based evaluation assesses the conceptual alignment between the generated ontology and its human-built counterpart. This ensures that the LLM-generated ontology retains meaningful relationships and terminology relevant to the domain. The semantic-based metrics include:

- Jaccard similarity (classes & relations), which computes the overlap between concepts and relations between ontologies using set-based similarity measures.
- Ontology embedding similarity, which applies Node2Vec to transform the ontologies into an embedding space and then compute cosine similarity to capture semantic coherence beyond exact term matching.
- Information content (IC), which measures how general or specific a concept is, with more specific concepts carrying higher IC values. Originaly derived from *Information Theory* [28], Resnik [29] introduced a corpus-based approach to computing IC, where the IC of a concept c is defined based on its probability of occurrence in a corpus as $-\log p(C)$, where $p(c)$ represents the probability of encountering concept c in a given corpus. Resnik's approach relies on external corpora to estimate concept probabilities, making it dependent on domain-specific datasets and prone to data sparsity issues. To overcome the dependency on external text corpora, Seco et al. [30] proposed an intrinsic IC measure that relies solely on the hierarchical structure of an ontology. Instead of using corpus frequency, their IC metric is based on the number of hyponyms (i.e., subsumed

concepts in the taxonomy), an adaption of the formula is shown below:

$$IC_(C) = 1 - \frac{\log(|\text{descendants}(C)| + 1)}{\log(|C|)} \quad (1)$$

Recent work [31, 27] has applied IC to ontology-based reasoning and explainability in AI.

These methods provide insights into how closely the generated ontologies align with human-built ones, capturing both direct classes and relations overlap and deeper latent relationships using embeddings.

Task-Based Methods: Task-based evaluation assesses the practical usability of the generated ontology by testing its effectiveness in augmenting tabular datasets for Machine Learning (ML) applications. The idea, introduced in [8, 9], is to leverage information from ontologies to enrich tabular data by adding additional features computed in the embedding space. These features capture latent semantic relationships between concepts, potentially improving predictive performance in downstream tasks. The process begins with populating the ontologies into knowledge graphs with tabular instances, associating data points with relevant ontology classes. Then the embeddings are computed. To systematically evaluate the impact of these ontology-based augmentations, we conduct experiments using four different ML models on a binary classification task. We assess model performance using accuracy and F2 score, with the latter being particularly relevant in disease prediction tasks, where maximizing true positive cases is critical for effectively identifying patients at risk. The evaluation considers two augmentation scenarios as follows:

- DistAugTab, which computes the euclidean distance between each instance embedding and the centroids of target classes (e.g., disease, no disease), enriching tabular data with proximity-based features.
- EmbedClustAygTab, which applies k-means clustering on the embeddings, incorporating cluster assignments and vector embeddings as additional features to improve dataset representation.

This evaluation aims to determine whether LLM-generated ontologies provide comparable results to human-built ones when used for tabular data augmentation, and whether certain embedding methods or ML models perform better with LLM-generated ontologies.

4. Experiment Setup

In this section we discuss the language models that are used to generate ontologies, followed by the details of the human-built ontology that served as gold standard.

LLMs used. To generate ontologies, we use three LLMs: GPT-4o[4], LLaMA 3.3[5], and DeepSeek R1[6], accessed via their respective APIs. These models were selected to provide a diverse comparison based on their training methodologies, their sizes and performance in structured knowledge generation tasks. Table 2 shows an overview of the LLMs that are used for the experiment, their version, size (in terms of parameters) and date of the conducted experiments.

Heart disease ontology. In the experiments, we compare the LLM-generated ontologies against a human-built ontology, (*Heart Disease Ontology*), to evaluate their structural and semantic quality. The Heart Disease Ontology, is a manually crafted ontology derived from Trepan Reloaded [27]. It

[4]https://openai.com/index/gpt-4o-system-card
[5]https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3
[6]https://api-docs.deepseek.com/news/news250120

Model	Version	Size/Parameters	Experiment Date
GPT-4o	GPT-4o-2024-11-20	Not disclosed	18/02/2025
LLaMa 3.3	Llama-3.3-70B-Instruct	70B	18/02/2025
DeepSeek R1	DeepSeek R1 [7]	685B	18/02/2025

Table 2
Overview of the LLMs used for the experiment, their versions, size, and date of the conducted experiments.

is designed to represent the key features found in the Heart Disease dataset [8], ensuring a structured and meaningful representation of relevant medical concepts. The ontology consists of 29 classes (e.g., Patient, HeartRate, ChestPain), 6 object properties (e.g., hasChestPain), and 10 data properties (e.g., hasAge), providing a well-defined knowledge structure for heart disease-related information.

Heart Disease dataset. The idea of generating ontologies from LLMs using concept seed, was to be able to represent tabular datasets' features. For our experiments, in order to evaluate the generated ontologies for task-based evaluation, we chose the two methods for augmenting the tabular data for ML binary classification, and for that the ontology needed to be populated into a knowledge graph with the instances from tabular data. For that we used Heart disease dataset in kaggle, which consists of 303 instances, with 14 features capturing various patient health indicators relevant to diagnosing heart disease such as heart rate and cholesterol.

5. Results

In this section, we present the evaluation results of the generated ontologies across structural-based, semantic-based and task-based. We analyze their structural characteristics, semantic similarity to human-built ontology and their impact on downstream tasks.

5.1. Structural-based results

Table 3 presents the structural properties of the generated ontologies compared to the human-built (HB) ontology. Starting with the maximum depth, which we already provided to the LLM to be maximum 5, we can see that only the ontology generated by DeepSeek has maximum depth of 5, whereas the other two LLMs generated a shallower ontologies with GPT-generated one being the shallowest with maximum depth of three.

The average degree, which represents the number of connections or relations per concept, is slightly lower in the LLM-generated ontologies (1.93) compared to HB (2.00). This indicates that LLMs generate slighly less connected structures.

Ontology	Structure/Graph-based				Semantic-based
	Max depth	Avg. degree	Path Length	Branch.	IC
HB	5	2.00	1.78	2.42	0.776
GPT-generated	3	1.93	1.20	4.00	0.868
LLaMA-generated	4	1.93	1.65	3.11	0.824
DeepSeek-generated	5	1.93	2.02	2.0	0.757

Table 3
Comparison of Human-Built and LLM-Generated Ontologies Based on Structural Characteristics.

A key difference is observed in the path length, which measures the average shortest distance between nodes/concepts. The GPT-generated ontology has the shortest path length (1.20), meaning concepts are

[8] https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

closer to each other and the root, reinforcing its flatter structure. LLaMa-generated ontology comes closer to HB in terms of the path length. In contrast, DeepSeek-generated (2.02) exceeds the HB ontology (1.78), suggesting that it organizes concepts in a more hierarchical manner, distributing them across multiple levels rather than clustering them near the root.

The branching factor, which indicates how widely concepts spread at each level, further supports these observations. The GPT-generated ontology has the highest branching factor (4.00), meaning it favors breadth over depth, potentially due to the model's tendency to generate broad taxonomies when not strictly constrained. On the other hand, DeepSeek-generated (2.00) has the lowest branching factor, aligning closely with the human-built ontology. This suggests that DeepSeek organizes concepts into deeper, more refined structures rather than spreading them widely.

Overall, these findings highlight key differences in the way LLMs structure ontologies. DeepSeek-generated ontology most closely resembles the HB ontology, balancing depth and path length, likely due to its ability to generate a more structured hierarchy. In contrast, GPT-generated ontology produces a much flatter structure with fewer hierarchical levels and wider branching, possibly reflecting a generative preference for more direct, less deeply nested relationships. Moreover, LLaMA-generated ontology represents a middle ground, generating a more structured hierarchy than GPT but still falling short of the depth and organization of the HB ontology.

Figure 2 presents the degree distribution of concepts across different ontologies, highlighting structural differences. The HB ontology shows a well-balanced hierarchy, with a mix of low-degree leaf concepts and high-degree core concepts. The GPT-generated ontology is highly skewed, with most concepts having low degrees (1-2) and a few high-degree nodes (12), leading to a broad but shallow structure. The LLaMA-generated ontology exhibits a more gradual degree distribution (1-7), balancing depth and breadth better than GPT while preserving hierarchical organization. The DeepSeek-generated ontology, while showing mostly low-degree nodes (1-4), aligns with its previously observed higher path length, suggesting that it prioritizes deeper hierarchical structuring over broad interconnectivity rather than being purely weakly connected. These patterns indicate that the HB and LLaMA-generated ontologies create more structured knowledge hierarchies, whereas GPT and DeepSeek-generated ontologies exhibit different structural biases—GPT towards flatter structures and DeepSeek towards deeper, more linear taxonomies.

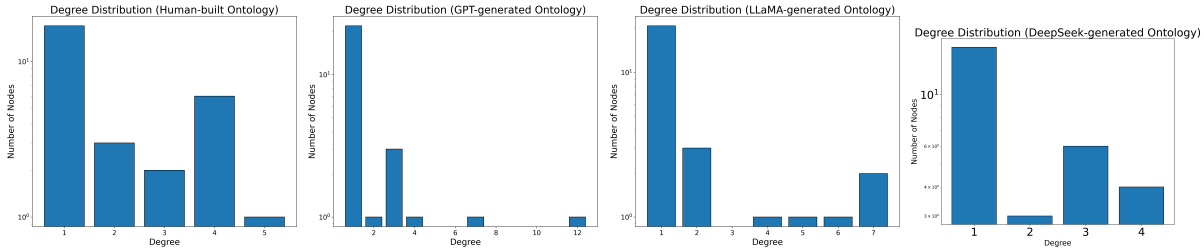

Figure 2: Degree distribution of ontologies: Human-created and LLM-generated ones.

5.2. Semantic-based results

The last column of Table 3 shows the average Information Content (IC) values for HB and LLM-generated ontologies. We observe that the GPT-generated ontology has the highest overall IC value (0.868), surpassing the human-built ontology (0.776). This suggests that GPT tends to generate concepts with higher specificity, potentially due to its tendency to introduce highly distinct categories, though in a flatter structure (as reflected in its low depth and high branching factor). However, this high IC does not necessarily translate to a well-organized hierarchy, as seen in its limited depth and high-degree generalizations.

The LLaMA-generated ontology (IC = 0.824) also exhibits a higher IC than the HB ontology. This aligns with its more structured hierarchy (moderate depth and balanced branching), suggesting that

LLaMA maintains a reasonable trade-off between depth and specificity, producing an ontology that is more hierarchical while still preserving semantic richness.

In contrast, the DeepSeek-generated ontology (IC = 0.757) has the lowest IC value. This suggests that DeepSeek produces a more generalized ontology with less concept differentiation, aligning with its deeper but less interconnected structure. The low branching factor (2.0) and longer path length (2.02) indicate that while it organizes concepts hierarchically, it does so with a more constrained level of semantic granularity, leading to lower IC.

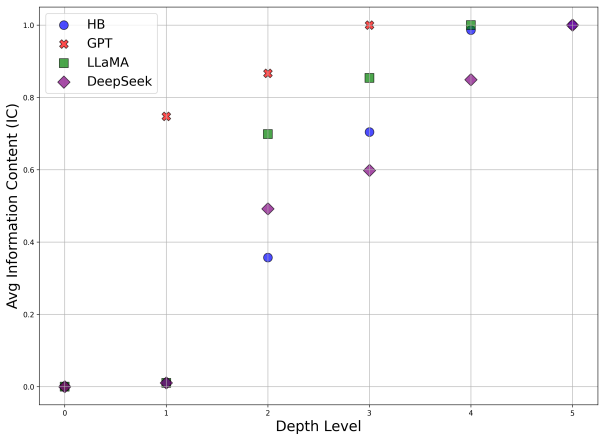

Figure 3: Comparison of IC at Different Depth Levels for Human and LLM Ontologies

Figure 3 further illustrates how IC varies across depth levels, showing that LLM-generated ontologies, particularly GPT and LLaMA, tend to reach higher IC values at mid-level depths, suggesting more specific categorizations even within shallow structures.

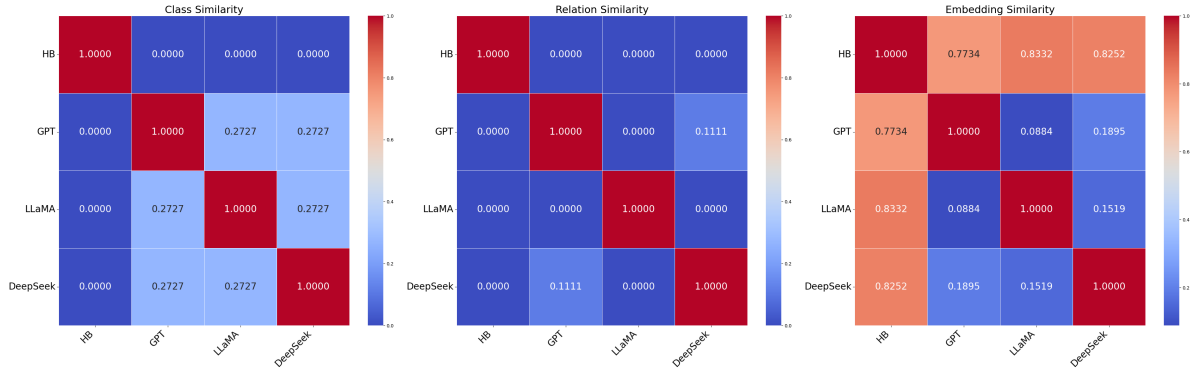

Figure 4: Comparison of ontology similarities using class structures, relations and embeddings across HB, GPT, LLaMA, and DeepSeek

Figure 4 shows the comparison of ontology similarities across the HB and LLM-generated ontologies. We observe that for Jaccard similarity, there is a huge difference between the LLM-generated ontologies compared to HB one. However, it is important to note that Jaccard similarity is a purely lexical metric and it does not account for synonyms or paraphrased labels. As a result, it may underestimate true semantic overlap when LLMs represent equivalent concepts using different terminology. Despite this, LLM-generated ontologies share some class similarities between each other, and this is due to the predefined terms provided during ontology construction. This suggests that while LLMs incorporate the given terms, their overall structural organization deviates from HB taxonomies.

In contrast, the Node2Vec-based cosine similarity of embeddings suggests a moderate alignment between the HB and LLM-generated ontologies, despite their structural differences. Notably, the HB ontology exhibits higher similarity to LLaMA (0.8332) and DeepSeek (0.8252) than to GPT (0.7734). This finding implies that although LLM-generated ontologies do not strictly replicate human-defined

structures, they capture meaningful semantic relationships in their embeddings. As a result, these ontologies may still hold value for downstream tasks such as knowledge graph completion and link prediction, where semantic coherence in the embedding space is crucial.

5.3. Task-based results

For task-based evaluation, we examine performance across two augmentation scenarios: DistAugTab and EmbedClustAugTab, which are further discussed in this subsection.

DistAugTab Scenario. Table 4 presents the accuracy and F2-score of various ML models trained on datasets augmented with features derived from measuring euclidean distance of each instance embedding to target class centroid. The centroid $\vec{c}_{C_j}$ is computed as the mean of the embedding vectors $\vec{v}_i$ for all instances belonging to the target class C_j. These embeddings are generated using different embedding methods. This evaluation assesses how effectively LLM-generated ontologies contribute to tabular data augmentation in downstream classification tasks.

Table 4
Performance Comparison of ML Models Across Ontologies Using Different Embedding Methods: Accuracy and F2 Score (Mean ± Std) in the DistAugTab Scenario

Model	**Accuracy (Mean ± Std)**				**F2 Score (Mean ± Std)**			
ML Model	Human	GPT	LLaMA	DeepSeek	Human	GPT	LLaMA	DeepSeek
Node2Vec								
KNN	**0.81** ± 0.04	0.70 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.70 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07
NN	**0.84** ± 0.08	0.81 ± 0.09	**0.84** ± 0.07	0.80 ± 0.12	**0.85** ± 0.05	0.83 ± 0.08	0.84 ± 0.08	0.80 ± 0.13
SVM	**0.82** ± 0.08	0.79 ± 0.10	**0.82** ± 0.08	0.79 ± 0.09	**0.82** ± 0.07	0.79 ± 0.09	**0.82** ± 0.07	0.79 ± 0.09
XGBoost	**0.82** ± 0.05	0.78 ± 0.08	0.81 ± 0.08	0.77 ± 0.10	**0.83** ± 0.06	0.78 ± 0.08	0.80 ± 0.07	0.77 ± 0.08
RDF2Vec								
KNN	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07
NN	0.80 ± 0.08	0.79 ± 0.11	**0.83** ± 0.06	0.79 ± 0.09	**0.82** ± 0.09	0.78 ± 0.11	**0.82** ± 0.08	0.77 ± 0.12
SVM	**0.80** ± 0.09	0.79 ± 0.10	**0.80** ± 0.09	0.79 ± 0.10	**0.81** ± 0.09	0.79 ± 0.10	0.80 ± 0.09	0.79 ± 0.10
XGBoost	**0.81** ± 0.07	0.78 ± 0.10	**0.81** ± 0.08	0.78 ± 0.10	**0.81** ± 0.07	0.77 ± 0.09	0.79 ± 0.09	0.74 ± 0.10
DistMult								
KNN	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.70 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07
NN	0.81 ± 0.08	0.79 ± 0.09	**0.82** ± 0.07	0.79 ± 0.11	**0.83** ± 0.07	0.80 ± 0.10	0.82 ± 0.07	0.81 ± 0.09
SVM	**0.81** ± 0.09	0.80 ± 0.10	**0.81** ± 0.09	0.79 ± 0.10	**0.81** ± 0.09	0.80 ± 0.10	**0.81** ± 0.09	0.80 ± 0.10
XGBoost	0.74 ± 0.14	**0.77** ± 0.07	**0.77** ± 0.08	**0.77** ± 0.07	0.74 ± 0.10	**0.77** ± 0.07	0.67 ± 0.14	0.76 ± 0.08
TransH								
KNN	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07
NN	**0.84** ± 0.07	0.82 ± 0.11	0.83 ± 0.06	0.81 ± 0.09	**0.85** ± 0.08	0.82 ± 0.10	0.84 ± 0.08	0.81 ± 0.11
SVM	**0.81** ± 0.09	0.79 ± 0.10	**0.81** ± 0.09	0.79 ± 0.10	**0.81** ± 0.09	0.79 ± 0.10	**0.81** ± 0.09	0.79 ± 0.10
XGBoost	0.76 ± 0.05	0.74 ± 0.07	**0.81** ± 0.07	0.77 ± 0.10	**0.79** ± 0.06	0.76 ± 0.08	0.78 ± 0.07	0.76 ± 0.08

Across all embedding methods (Node2Vec, RDF2Vec, DistMult and TransH), as expected, HB ontology consistently achieves the highest or near-highest performance across both accuracy and F2-score.

Among the LLM-generated ontologies, the LLaMA-generated ontology generally outperforms GPT and DeepSeek, with DeepSeek consistently ranking the lowest. While LLaMA-generated ontology comes closest to the HB ontology, making it the most suitable for downstream ML tasks, the GPT-generated ontology exhibits inconsistent performance, with lower overall accuracy and F2-score, likely due to its flatter structure and lack of deeper hierarchical relationships. The DeepSeek-generated ontology frequently achieves the lowest performance, particularly when DistMult and TransH are used to create embeddings, suggesting that its structural and semantic characteristics, as well as the embedding vectors do not translate effectively into tabular data augmentation.

Interestingly, the XGBoost model, which typically performs well in tabular classification tasks, shows greater variability in performance across different ontologies, indicating that it is more sensitive to

ontology quality than other ML models. Notably, when embeddings are generated using DistMult, XGBoost achieves higher accuracy and F2-score with LLM-generated ontologies, especially GPT-generated, compared to the HB ontology. This suggests that in certain cases, LLM-generated ontologies may capture latent semantic relationships that benefit specific ML models, despite their structural limitations.

EmbedClustAugTab Scenario. Table 5 shows the accuracy and F2 score of various ML models trained on datasets augmented using cluster-based features derived from embeddings. Unlike the DistAugTab scenario, where features were computed based on Euclidean distances to target class centroids, the EmbedClustAugTab scenario enriches the dataset by incorporating K-means cluster assignments and embedding vectors for each instance. This evaluation analyzes whether LLM-generated ontologies provide useful representations for downstream classification tasks when cluster-based features are used instead of distance-based features.

Table 5
Performance Comparison of ML Models Across Ontologies Using Different Embedding Methods: Accuracy and F2 Score (Mean ± Std) in the EmbedClustAugTab Scenario

Model	**Accuracy (Mean ± Std)**				**F2 Score (Mean ± Std)**			
ML Model	Human	GPT	LLaMA	DeepSeek	Human	GPT	LLaMA	DeepSeek
Node2Vec								
KNN	**0.82** ± 0.04	0.71 ± 0.06	0.81 ± 0.04	0.70 ± 0.07	**0.81** ± 0.04	0.70 ± 0.07	**0.81** ± 0.04	0.70 ± 0.08
NN	**0.83** ± 0.07	0.80 ± 0.12	0.82 ± 0.08	0.77 ± 0.15	**0.82** ± 0.08	0.79 ± 0.11	**0.82** ± 0.08	0.80 ± 0.09
SVM	**0.81** ± 0.09	0.80 ± 0.10	**0.81** ± 0.07	0.79 ± 0.08	**0.82** ± 0.07	0.80 ± 0.09	0.81 ± 0.07	0.80 ± 0.08
XGBoost	**0.79** ± 0.07	0.76 ± 0.09	0.77 ± 0.05	0.73 ± 0.07	0.77 ± 0.07	0.73 ± 0.10	**0.78** ± 0.05	0.75 ± 0.10
RDF2Vec								
KNN	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.71 ± 0.07	0.81 ± 0.04	0.70 ± 0.06	0.81 ± 0.04	0.71 ± 0.07
NN	0.81 ± 0.09	0.78 ± 0.13	**0.84** ± 0.06	0.80 ± 0.10	**0.81** ± 0.08	0.77 ± 0.12	**0.84** ± 0.08	**0.81** ± 0.12
SVM	**0.81** ± 0.09	0.79 ± 0.10	**0.81** ± 0.08	0.79 ± 0.09	**0.81** ± 0.08	0.78 ± 0.09	**0.81** ± 0.07	0.80 ± 0.08
XGBoost	0.78 ± 0.03	0.74 ± 0.07	**0.80** ± 0.06	0.74 ± 0.09	**0.79** ± 0.07	0.75 ± 0.09	0.76 ± 0.09	0.74 ± 0.09
DistMult								
KNN	**0.81** ± 0.04	0.71 ± 0.06	**0.81** ± 0.04	0.70 ± 0.07	**0.81** ± 0.05	0.70 ± 0.07	**0.81** ± 0.04	0.71 ± 0.06
NN	0.81 ± 0.08	0.79 ± 0.11	0.82 ± 0.08	0.78 ± 0.13	0.83 ± 0.08	**0.82** ± 0.11	0.81 ± 0.08	0.78 ± 0.13
SVM	**0.81** ± 0.09	0.79 ± 0.10	**0.81** ± 0.09	0.78 ± 0.11	**0.81** ± 0.09	0.80 ± 0.11	0.80 ± 0.09	0.79 ± 0.11
XGBoost	0.74 ± 0.10	**0.76** ± 0.11	0.70 ± 0.09	0.66 ± 0.23	**0.73** ± 0.07	0.71 ± 0.17	0.67 ± 0.09	0.66 ± 0.19
TransH								
KNN	**0.81** ± 0.04	0.71 ± 0.06	**0.81** ± 0.04	0.70 ± 0.06	**0.81** ± 0.04	0.71 ± 0.07	**0.81** ± 0.04	0.70 ± 0.07
NN	**0.82** ± 0.08	0.78 ± 0.13	**0.82** ± 0.07	0.81 ± 0.08	0.81 ± 0.07	0.76 ± 0.09	**0.82** ± 0.06	0.79 ± 0.10
SVM	0.80 ± 0.10	0.78 ± 0.12	**0.81** ± 0.09	0.80 ± 0.10	0.81 ± 0.09	0.78 ± 0.10	**0.82** ± 0.09	0.79 ± 0.11
XGBoost	0.70 ± 0.13	0.74 ± 0.12	**0.75** ± 0.07	0.72 ± 0.12	0.73 ± 0.09	**0.76** ± 0.08	0.74 ± 0.07	0.68 ± 0.14

When comparing the two augmentation approaches, performance gaps between HB and LLM-generated ontologies appear more pronounced in the EmbedClustAugTab scenario, especially when the embeddings are generated using TransH. This suggests that distance-based features (DistAugTab) are more robust to variations in ontology structure, while cluster-based features (EmbedClustAugTab) are more sensitive to how well an ontology organizes and differentiates concepts. As a result, models in this scenario rely more heavily on ontology quality to provide meaningful cluster assignments, highlighting the importance of well-structured taxonomies.

Regarding ML models, NNs and SVMs generally perform better across embeddings, confirming that they benefit more from additional embedding-based features. XGBoost continues to exhibit higher sensitivity to ontology quality, with noticeable performance variations across ontologies and embedding methods. KNN performs relatively stably across different embedding methods but shows stronger dependence on ontology structure, performing best with HB and LLaMA-generated ontologies.

When analyzing embedding methods, Node2Vec and RDF2Vec embeddings continue to yield the highest accuracy and F2-score, consistent with the DistAugTab scenario. DistMult and TransH embed-

dings perform worse, particularly when used to generated embeddings for LLM-generated ontologies. Additionally, the performance gap between HB and LLM-generated ontologies is wider in this scenario, suggesting that clustering-based augmentation methods are more reliant on well-structured ontologies than distance-based methods.

In conclusion, HB ontologies still lead in performance, confirming that human-built ontologies provide the best augmentation for ML tasks. LLaMA-generated ontologies remain the best LLM-generated alternative, showing stronger alignment with HB than GPT and DeepSeek.

6. Conclusion and Future Work

In this paper, we present an evaluation strategy for ontologies generated with LLMs. We investigate the potential of LLMs to generate domain-specific ontologies, given a set of concepts and characteristics, in the domain of heart disease. We evaluated the LLM-generated ontologies using structural, semantic and task-based metrics.

For RQ1, the results indicate that LLM-generated ontologies partially align with human-built ones but deviate in hierarchical organization and conceptual relationships. Structurally, the ontology generated using LLaMA has more structured knowledge hierarchies, whereas GPT and DeepSeek-generated ontologies exhibit different structural biases—GPT towards flatter structures and DeepSeek towards deeper, more linear taxonomies. Semantically, they fail to maintain the same classes and relations as the human based ontology, as reflected in low Jaccard similarity, however they show moderate Node2Vec-based cosine similarity, especially the ontology generated using LLaMa. This suggests that although they do not closely resemble human-built ontologies structurally, they can still be valuable in embedding-based applications such as knowledge graph completion and link prediction, where structural precision is less critical than semantic coherence. In task-based evaluation, LLaMA-generated ontologies perform the best among other LLMs, suggesting that some LLMs may produce useful ontologies for dataset augmentation. Notably, LLMs can be effectively used to generate ontologies when the primary goal is to model dataset features (given as a set of terms) or to enforce specific structural constraints, such as defining the number of classes, relations, or maximum depth.

Regarding RQ2, we evaluate LLM-generated ontologies through structural, semantic, and task-based evaluations. Graph-theoretic metrics (depth, degree, path length) evaluate hierarchical organization, while Jaccard similarity and Node2Vec-based cosine similarity measure semantic alignment. In task-based evaluation, ontologies are converted into knowledge graphs, embedded using Node2Vec, RDF2Vec, DistMult, and TransH, and used for tabular data augmentation in ML models. Performance is tested in two augmentation scenarios: DistAugTab and EmbedClustAugTab.

For RQ3, evaluation shows that LLaMA-generated ontology is the closest to the human-built ontology, producing more structured and semantically meaningful ontologies. It achieves deeper hierarchies, better preserves conceptual relationships, and consistently outperforms GPT and DeepSeek in task-based evaluation. However, it still does not fully replicate human-built ontologies, especially in taxonomic structuring and relation modeling. GPT-generated ontologies are inconsistent, favoring broad, flat structures, while DeepSeek struggles the most, generating weakly interconnected hierarchies with poor downstream usability.

Limitations. While this paper provides valuable insights, its focus on a single domain limits the generalizability of the findings to other domains. Moreover, potential risks such as LLM training data leakage and sensitivity to prompt phrasing were not investigated, though they may influence the quality and reliability of the generated ontologies.

Future work. Future research will explore Competency Question (CQ) evaluation to assess how well LLM-generated ontologies support domain-specific queries, as well as schema validation using SHACL (Shapes Constraint Language) will be incorporated to ensure structural and consistency checks, providing a more rigorous evaluation of ontology quality. Additionally, we plan to extend the study beyond the medical domain to evaluate LLM performance in other domains. Further experiments will include a broader range of LLMs and investigate their effectiveness in generating larger ontologies,

incorporating techniques such as ontology modularization to improve scalability. We will also assess alternative task-based evaluation approaches to better understand the practical usability of AI-generated ontologies.

7. Acknowledgments

This work was supported by the FFG SENSE (894802) and FAIR-AI (904624) projects, as well as by the Austrian Science Fund (FWF) BILAI 10.55776/COE12 and HOnEst (V 745-N) projects. For open access purposes, the author has applied a CC BY public copyright license to any author accepted manuscript version arising from this submission.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4o in order to: brainstorm ideas about the title, as well as code refactoring. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] H. Babaei Giglou, J. D'Souza, S. Auer, Llms4ol: Large language models for ontology learning, in: International Semantic Web Conference, Springer, 2023, pp. 408–427.
[2] R. Confalonieri, G. Guizzardi, On the multiple roles of ontologies in explanations for neuro-symbolic ai, Neurosymbolic Artificial Intelligence (2024) 1–15.
[3] N. Hubert, P. Monnin, M. d'Aquin, D. Monticolo, A. Brun, Pygraft: Configurable generation of synthetic schemas and knowledge graphs at your fingertips, in: European Semantic Web Conference, Springer, 2024, pp. 3–20.
[4] A. Melo, H. Paulheim, Synthesizing knowledge graphs for link and type prediction benchmarking, in: The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14, Springer, 2017, pp. 136–151.
[5] D. Doumanas, A. Soularidis, D. Spiliotopoulos, C. Vassilakis, K. Kotis, Fine-tuning large language models for ontology engineering: A comparative analysis of gpt-4 and mistral, Applied Sciences 15 (2025) 2146.
[6] R. Alharbi, V. Tamma, F. Grasso, T. R. Payne, Investigating open source llms to retrofit competency questions in ontology engineering, in: Proceedings of the AAAI Symposium Series, volume 4, 2024, pp. 188–198.
[7] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, arXiv preprint arXiv:2403.08345 (2024).
[8] M. Llugiqi, F. J. Ekaputra, M. Sabou, Enhancing machine learning predictions through knowledge graph embeddings, in: International Conference on Neural-Symbolic Learning and Reasoning, Springer, 2024, pp. 279–295.
[9] M. Llugiqi, F. J. Ekaputra, M. Sabou, Semantic-based data augmentation for machine learning prediction enhancement, Neurosymbolic Artificial Intelligence (under review) (2025).
[10] D. Kontokostas, A. Zaveri, S. Auer, J. Lehmann, Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data, in: Knowledge Engineering and the Semantic Web: 4th International Conference, KESW 2013, St. Petersburg, Russia, October 7-9, 2013. Proceedings 4, Springer, 2013, pp. 265–272.
[11] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, Hermit: an owl 2 reasoner, Journal of automated reasoning 53 (2014) 245–269.
[12] N. Guarino, C. A. Welty, An overview of ontoclean, Handbook on ontologies (2009) 201–220.
[13] S. Tsaneva, S. Vasic, M. Sabou, Llm-driven ontology evaluation: Verifying ontology restrictions with chatgpt, The Semantic Web: ESWC Satellite Events 2024 (2024).

[14] M. Kejriwal, C. A. Knoblock, P. Szekely, Knowledge graphs: Fundamentals, techniques, and applications, MIT Press, 2021.
[15] J. Raad, C. Cruz, A survey on ontology evaluation methods, in: International conference on knowledge engineering and ontology development, volume 2, SciTePress, 2015, pp. 179–186.
[16] R. Alharbi, V. Tamma, F. Grasso, T. Payne, An experiment in retrofitting competency questions for existing ontologies, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, 2024, pp. 1650–1658.
[17] Y. Rebboud, L. Tailhardat, P. Lisena, R. Troncy, Can llms generate competency questions?, in: European Semantic Web Conference, Springer, 2024, pp. 71–80.
[18] P. Mateiu, A. Groza, Ontology engineering with large language models, in: 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, 2023, pp. 226–229.
[19] N. Fathallah, A. Das, S. D. Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, Neon-gpt: a large language model-powered pipeline for ontology learning, in: European Semantic Web Conference, Springer, 2024, pp. 36–50.
[20] A. S. Lippolis, M. Ceriani, S. Zuppiroli, A. G. Nuzzolese, Ontogenia: Ontology generation with metacognitive prompting in large language models, in: European Semantic Web Conference, Springer, 2024, pp. 259–265.
[21] M. J. Saeedizade, E. Blomqvist, Navigating ontology development with large language models, in: European Semantic Web Conference, Springer, 2024, pp. 143–161.
[22] Y. Zhang, Z. Chen, L. Guo, Y. Xu, W. Zhang, H. Chen, Making large language models perform better in knowledge graph completion, in: Proceedings of the 32nd ACM International Conference on Multimedia, 2024, pp. 233–242.
[23] X. Li, A. J. Hughes, M. Llugiqi, F. Polat, P. Groth, F. J. Ekaputra, et al., Knowledge-centric prompt composition for knowledge base construction from pre-trained language models., in: KBC-LM/LM-KBC@ ISWC, 2023.
[24] H. Babaei Giglou, J. D'Souza, F. Engel, S. Auer, Llms4om: Matching ontologies with large language models, in: European Semantic Web Conference, Springer, 2024, pp. 25–35.
[25] C. Saetia, J. Phruetthiset, T. Chalothorn, M. Lertsutthiwong, S. Taerungruang, P. Buabthong, Financial product ontology population with large language models, in: Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing, 2024, pp. 53–60.
[26] X. Wu, Y.-L. Li, J. Sun, C. Lu, Symbol-llm: leverage language models for symbolic system in visual human activity reasoning, Advances in Neural Information Processing Systems 36 (2023) 29680–29691.
[27] R. Confalonieri, T. Weyde, T. R. Besold, F. M. del Prado Martín, Using ontologies to enhance human understandability of global post-hoc explanations of black-box models, Artificial Intelligence 296 (2021) 103471.
[28] S. Ross, First Course in Probability, A, Macmillan, 1976.
[29] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, arXiv preprint cmp-lg/9511007 (1995).
[30] N. Seco, T. Veale, J. Hayes, An intrinsic information content metric for semantic similarity in wordnet, in: Ecai, volume 16, 2004, p. 1089.
[31] D. Sánchez, M. Batet, D. Isern, Ontology-based information content computation, Knowledge-based systems 24 (2011) 297–303.

A. Online Resources

The generated ontologies are available in

- Zenodo