Generating Ontology-Learning Training-Data through Verbalization

Antonio Zaitoun[1], **Tomer Sagi**[2], **Mor Peleg**[1]

[1]University of Haifa, Israel
[2]Aalborg University, Denmark
azaitoun@campus.haifa.ac.il, morpeleg@is.haifa.ac.il, tsagi@cs.aau.dk

Abstract

Ontologies play an important role in the organization and representation of knowledge. However, in most cases, ontologies do not fully cover domain knowledge, resulting in a gap. This gap, often expressed as a lack of concepts, relations, or axioms, is usually filled by domain experts in a manual and tedious process. Utilizing large language models (LLMs) can ease this process; a fine-tuned LLM could receive as input up-to-date and reliable domain knowledge natural text and output a structured graph in OWL RDF/Turtle format, which is the standard format of ontologies. Thus, to fine-tune a model, text-owl sentence pairs that constitute such a dataset must be acquired. Unfortunately, such a dataset does not exist in the literature or within the open-source community. Therefore, this paper introduces our LLM-assisted verbalizer to create such a data set by converting OWL statements from existing ontologies into natural text. We evaluate the verbalizer on 322 classes from four different ontologies using two different LLMs, achieving precision and recall as high as 0.99 and 0.96, respectively.

Introduction

Gruber (Gruber 1995) defines an ontology as an explicit specification of a conceptualization. Traditionally, it comprises concepts, each characterized by multiple attributes and relationships that tie them together. Moreover, an ontology can encompass individuals, which are instances of specific concepts and may include axioms that define logical constraints or assertions. Thus, ontologies describe a single domain and are employed to abstract and formally capture the semantic meaning behind the concepts and their interrelations.

There is increasing interest in utilizing ontologies to enable semantic reasoning capabilities. For instance, ontologies play a crucial role in data integration by unifying and relating data elements under concepts despite differing schemas (Mountasser et al. 2021). Ontologies are also utilized in software engineering (Zada et al. 2023), requirement engineering (Yang, Cormican, and Yu 2019), data retrieval (Taglino et al. 2023), and decision support (Peleg et al. 2024). Ontologies are applied across diverse industries, including oil and gas, military, e-government, e-commerce, and healthcare (Asim et al. 2018).

Despite the significant benefits of ontologies, their development and maintenance are challenging. The primary challenge lies in the manual efforts required for their creation and maintenance (Alobaidi, Malik, and Sabra 2018). Consequently, due to the labor-intensive nature of maintaining ontologies, they often lag behind in reflecting the latest developments (del Valle et al. 2019). For instance, a biomedical ontology might miss newly developed medications, immunizations, or discovered pathogens.

Current approaches to ontology learning leverage the capabilities of large language models (LLMs) in tasks such as concept extraction (Dunn et al. 2022) and hierarchical relationship extraction (is-a relationships). However, these methods face limitations. Non-hierarchical relationship extraction, for instance, shows only moderate success (Giglou, D'Souza, and Auer 2023), and axiom extraction remains largely unexplored (Watrobski 2020). Furthermore, some studies indicate that LLMs may perform poorly in relationship extraction tasks when not specifically trained for such purposes (Giglou, D'Souza, and Auer 2023; Zaitoun et al. 2023).

Given the advanced capabilities of LLMs, it is imperative to harness these models effectively for tasks like concept, relationship, and axiom extraction. Our long-term goal is to utilize LLMs to extract knowledge from trusted sources of unstructured textual data, such as meta-reviews and clinical guidelines, to identify new concepts, relationships, and axioms. However, existing LLMs are not specifically trained for axiom extraction (Zaitoun 2024a). Potentially, one could fine-tune LLMs for the task of ontology fragment generation. By providing the model with a dataset comprising OWL (Web Ontology Language 2012) fragment and text-equivalent pairs, a fine-tuned model could predict OWL statements from natural language text inputs.

However, since such a dataset is not publicly available, we present a method to reliably generate such datasets from existing ontologies. The proposed method utilizes an LLM-assisted verbalizer to generate natural language text for ontology fragments, enabling the construction of an end-to-end AI pipeline (Figure 1). This pipeline begins by gathering ontologies, verbalizing them into natural language statements, and using these statements to fine-tune existing foun-

dational models for OWL generation. In this paper, we focus on developing and evaluating the verbalizer, more specifically, the methodology for the dataset generation phase of the pipeline.

Related Work

Over the years, numerous Controlled Natural Languages (CNLs) have been introduced. CNL, as defined by (Kuhn 2014), is a constructed language based on a specific natural language characterized by a more restrictive lexicon, syntax, and/or semantics while preserving most of its natural properties. One popular example of a CNL for representing structured knowledge using English text is Attempt-Controlled-English (ACE) (Kaljurand 2007). These languages facilitate interactions with formal ontological statements, making them more accessible and faster for users unfamiliar with formal notations.

NaturalOWL (Androutsopoulos, Lampouras, and Galanis 2013) is a pioneer in the field of OWL verbalization, applying Natural Language Generation (NLG) techniques to convert OWL ontologies into natural language statements using templates and sentence aggregations beyond CNLs. However, their solution is built on rigid rules that assume specific content within the ontology, limiting the expressiveness and flexibility of the generated descriptions. In contrast, our approach leverages the adaptability of LLMs, allowing for more nuanced and context-sensitive verbalizations that aren't confined by such rigid assumptions.

Similarly, Mille et al. (Mille, Dasiopoulou, and Wanner 2019) implemented a domain-specific template-based system for the verbalization of semantic web datasets. While template-based systems are highly reliable, they suffer from low portability since new templates must be created for each new domain, style, or language. Furthermore, this approach addresses the verbalization of datasets rather than OWL ontologies. Our LLM-based approach, on the other hand, is domain-agnostic and adaptable, providing a more flexible solution that doesn't require extensive reconfiguration for new domains.

Alternatively, (He et al. 2023) implemented a recursive pattern-based verbalizer that produces CNL statements with the intent of analyzing the knowledge coverage of pre-trained LLMs. While this verbalizer shows promise with support for complex expressions and zero configuration requirements, it lacks comprehensive support for certain OWL notations (e.g., *UnionOf*, *ObjectIntersectionOf*, *ObjectMinCardinality*), and has limited label pre-processing—using the full IRI hen a class lacks the *rdfs:label* tag. For instance, *http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl#hasTopping* is used instead of *"has topping"*. Our approach, however, provides a broader and extendable OWL notation support and more robust handling of ontology labels, ensuring more accurate and natural-sounding verbalizations.

Vidange et al. (Vidanage et al. 2021) adopted an AI-based approach for ontology verbalization using a chatbot, specifically Google's AliceBot (Wallace 2009), which utilizes the XML schema known as Artificial Intelligence Markup Language (AIML) to specify conversation rules. Their goal was to verbalize ontology content to generate a user guide detailing changes or differences in a version. Although their work addresses the limitations of traditional CNLs by moving beyond controlled versions of English, AliceBot is limited in variability and lacks the advanced AI reasoning capabilities found in LLMs. Our approach, by contrast, harnesses the full power of LLMs, offering more dynamic and context-aware verbalizations that can better capture the nuances of ontology content.

In summary, unlike the aforementioned approaches, we present here an LLM-assisted, OWL-specific verbalizer with pattern support for extendability that is domain-agnostic, allowing for the verbalization of ontologies with minimal effort. This approach combines the flexibility and adaptability of LLMs with robust OWL support, providing a more versatile and comprehensive solution for ontology verbalization.

Method

This section delineates our methodology for generating textual descriptions of OWL classes utilizing our LLM-assisted verbalizer. The verbalizer operates through three primary stages: *Initialization*, *CNL sentence Generation*, and *Paraphrasing*.

During the *initialization* phase, a vocabulary of concepts and relationships is constructed by querying all classes and relationships from the ontology. The textual labels that are assigned to classes and relationships. Once the vocabulary is established, verbalization is performed given a concept identifier as input. The verbalizer receives a single concept identifier (IRI) and recursively traverses related concepts and axioms, resulting in a tree with the initial concept as the root and the discovered concepts as leaves. Next, *CNL sentences are generated* for each unique path from the root to each leaf. A configurable pattern mechanism detects specific patterns in the OWL W3C standard to consolidate the sentences into more coherent ones. While these sentences describe the OWL fragment's contents and are readable, they are often grammatically incorrect and do not resemble naturally written text. Therefore, an additional step involves employing a general-purpose LLM to *paraphrase* the sentences into coherent paragraphs. Figure 2 depicts this process. We experimented with two LLMs for paraphrasing during the evaluation of the verbalizer: GPT-4o, using OpenAI's official API (Achiam et al. 2023), and Llama 3 (8B) (Meta 2024), running locally using Ollama.

Initialization

During the verbalizer's initialization, settings such as classes or relationships to ignore, and the ability to rephrase or override the labels of certain relationships can be configured. This helps eliminate noise from relationships containing metadata and allows renaming relationships for more coherent sentences. For instance, the relationship *hasDbXref* is ignored due to its lack of beneficial value to the verbalized sentence. This is because its presence does not contribute to producing any naturally sounding sentences that are likely to be observed in text from domain knowledge (as seen in (Zaitoun, Sagi, and Peleg 2024)). Furthermore, axioms such

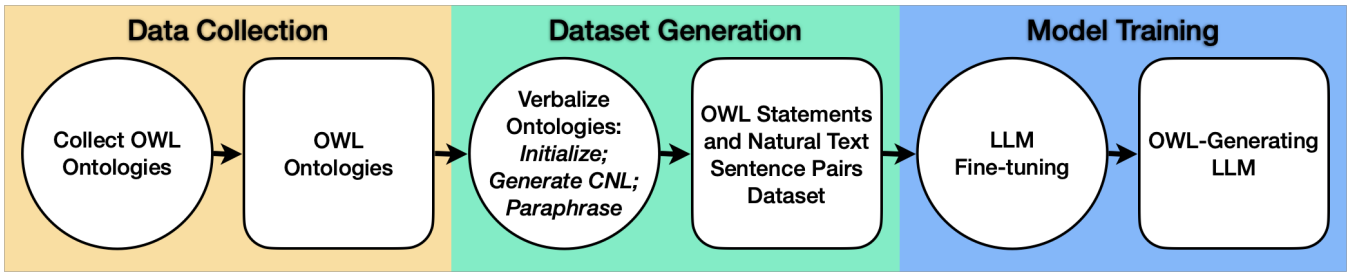


Figure 1: Proposed end-to-end AI pipeline comprised of three stages: 1) Data collection, 2) Dataset Generation, 3) Model Training. Circles represent processes or actions while squares represent artifacts and outputs.

as *owl:subClassOf* and *owl:equivalentClass* are rendered as "is a type of" and "is the same as," to make the text more natural. These adjustments significantly impact the resulting sentences when they are later paraphrased resulting in more natural sentences and paragraphs.

Once these parameters are specified, the vocabulary is automatically constructed by querying all the classes and relationships, generating a lookup table with the identifier as the key and the label as the value. Labels are obtained from the *rdfs:label* property or, alternatively, using the entity's identifier, which is tokenized based on capitalization. For example, the relationship *hasTopping* becomes *has topping*.

Pattern Configuration

Our verbalizer includes a mechanism for writing custom patterns used during concept traversal. These patterns enable the customization of the generated tree, enhancing the verbalized sentences. For example, when defining relationships between classes, the Restriction clause may be used with two properties: *onProperty* and *someValuesFrom*. Without a pattern, this structure results in two distinct sentences.

Consider the following example of *Interesting Pizza*. An interesting pizza is any pizza that has at least three toppings.

```
:interesting_pizza a owl:Class ;
  owl:equivalentClass [
    a owl:Class ;
    owl:intersectionOf [
       rdf:first [
          a owl:Restriction ;
          owl:minCardinality "3"^^xsd:int ;
          owl:onProperty :has_topping
       ] ;
       rdf:rest [
          rdf:first :pizza ;
          rdf:rest :nil
       ]
    ]
] .
```

Without any patterns, the CNL statement would be generated as:

```
interesting pizza is same as all of (first (min
     cardinality 3, and on property a has topping),
     and rest (first a pizza, and rest a nil)).
```

To address this, we have implemented three specific patterns for OWL to normalize the generated tree into a more coherent structure:

1 **Restriction pattern** - This pattern takes care of any restriction, properties, and modifiers alike, including cardinality.

2 **List pattern** - Because OWL is implemented using triples, a common design pattern for managing ordered sets is the *first-rest* pattern. When there is a list, a starting node has two nodes connected to it via two relationships *rdf:first* and *rdf:rest*. This continues until one of the nodes points to *:nil*, indicating the list's end. The purpose of this pattern is to simplify this structure and aggregate the values into a list.

3 **Disjointness pattern** - In many ontologies, due to the open-world reasoning, ontology designers choose to use disjoint, to indicate an instance cannot be of both classes. This results in many disjoints, which result in many sentences during verbalization. This pattern groups all disjoint statements into a single statement.

With the patterns in place, the generated CNL looks as follows:

```
interesting pizza is same as all of (has at least
     3 toppings, and a pizza).
```

CNL Sentence Generation

The verbalization process begins with a concept (class) identifier as input. The verbalizer queries the ontology to resolve all first-degree related concepts and constructs a tree, preserving relationship information. This process is recursively performed for each leaf until no further concepts are added. Before adding more child nodes, a check is conducted using the configured patterns, with only one pattern evaluated per set of results. When a pattern match occurs, a normalization step takes place instead of merely appending child nodes to the tree. Once the tree is constructed, a recursive algorithm traverses each unique path to form a single CNL statement. The output of our verbalizer is similar to that of ACE (Kaljurand 2007) and the OWL Manchester Syntax (Horridge et al. 2006), both of which are well-established standards closely related to CNL. However, we opted to implement our own custom CNL output. This is because these CNLs are not

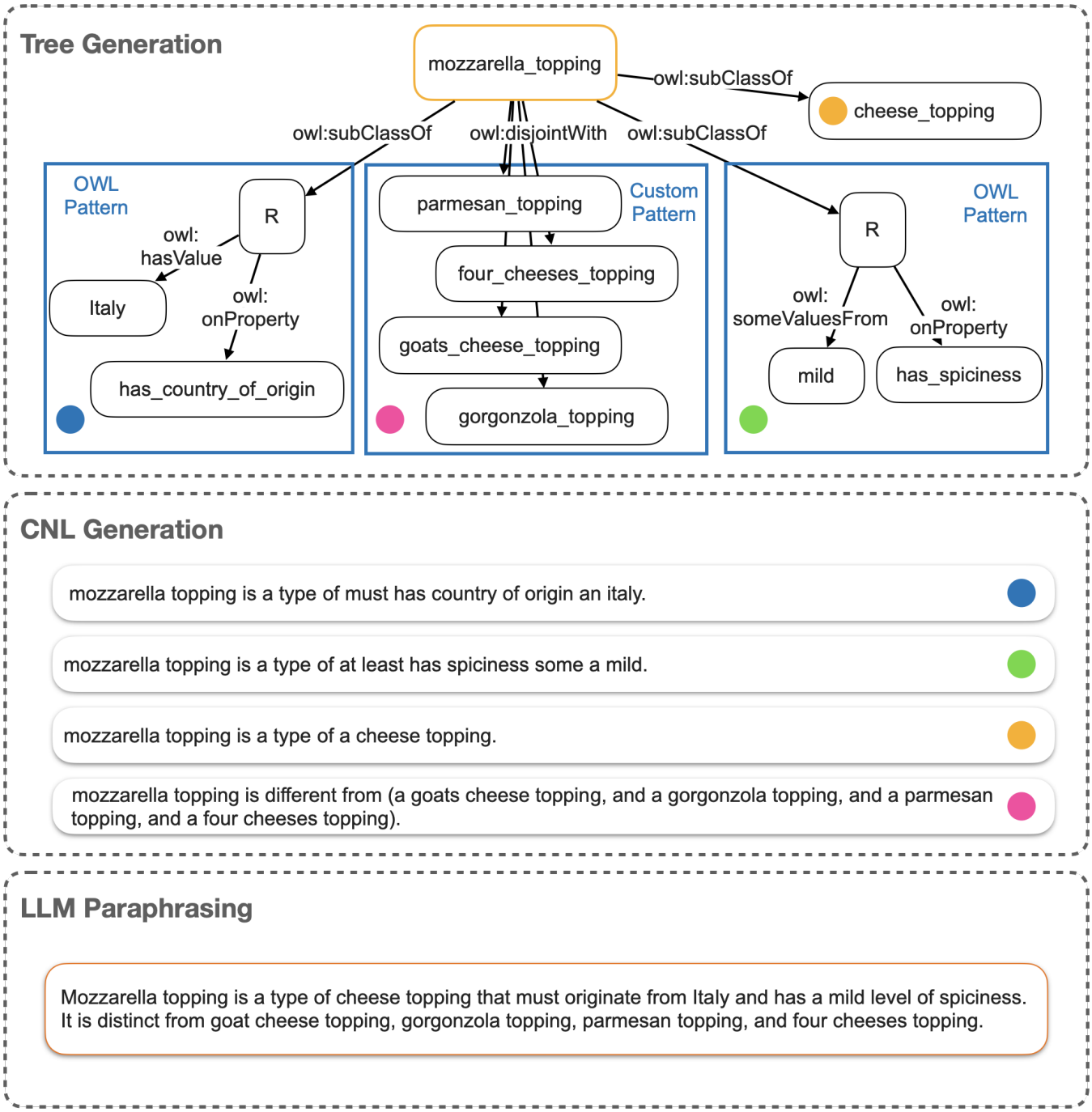


Figure 2: The verbalization process. In this example, the input concept is *Mozzarella Topping*, which is then traversed to discover neighboring concepts. These concepts are added to the tree with the respected relationship. Based on pre-defined patterns, some nodes are re-arranged to result in a single path. Each path is then converted into a CNL statement. These statements are then fed into an LLM with a corresponding prompt to generate a cohesive sentence or paragraph.

customizable, more specifically, the Manchester syntax is too close the OWL notation making it less natural, e.g., *MargheritaPizza subClassOf Pizza*, whereas we preferred the natural sentence *Margherita Pizza is a type of Pizza*. This choice offers greater flexibility in sentence generation, allowing the output to be not only OWL-compatible but also configurable for various use cases. In addition, all the original triples used in the tree generation process are collected and persevered to allow the reconstruction of the original OWL statements. The full algorithm is described online (Zaitoun 2024b).

LLM Paraphrasing

While CNL statements generated syntactically are readable, similar to ACE (Fuchs, Kaljurand, and Kuhn 2008), they do not resemble naturally written text. Thus, the paraphrasing step is introduced. In this step, we leverage existing LLMs and a method called prompt engineering (Marvin et al. 2023), to prompt the LLM to rewrite the sentences without losing their meaning. This ensures that they are grammatically correct and coherent. We propose the following prompt template:

```
1  You are an extremely specific data expert capable
       of converting pseudo English sentences into a
       meaningful and casual paragraph without losing
       information. Avoid repeating information.
       Spell out everything, don't be lazy!
2  ...
3
4  {content}
```

This results in the below output, which is far more natural:

```
1  An interesting pizza is one that has at least
       three toppings and is, of course, a pizza.
```

Empirical Evaluation

In the following section, we detail our experimental evaluation. We begin by describing the ontologies used in the evaluation and then describe the evaluation method used to quantify the verbalization quality.

Datasets

We evaluated our approach on four ontologies (Table 1), of which two (Pizza Ontology, People Ontology) are example ontologies created for educational purposes and used in tutorials such as the Protege user guide (Horridge et al. 2004), and two are samples from real-world ontologies created for scientific domain representation. A sample of 107 classes and their related constructs from the Foundational Model of Anatomy (FMA) (Rosse and Mejino Jr 2003), containing 104,721 classes and 168 property types, and a sample of 100 classes from the Mondo Disease Ontology (MONDO) (Vasilevsky et al. 2022).

Evaluation Methods

We performed the paraphrasing step of the Verbalizer with two LLMs (GPT-4o and Llama 3) for each of the four ontologies. We then manually evaluated the quality of the resulting text. Thus, the focus of this evaluation, is the performance of LLMs on the tasks of CNL generation and paraphrasing. More specifically, we wanted to ensure the generated text does not omit any details provided in the CNL statements, nor does it introduce additional information (even if it was correct). Therefore, for each verbalized concept, we evaluated the correctly paraphrased statements as True Positive (TP), missed statements as False Negative (FN), and extra statements or additional information as False Positive (FP). One author conducted the evaluation across all four ontologies for the two models, while the other two authors reviewed 10% of diverse samples to ensure agreement and consistency. In cases of disagreement, discussions were held to reach alignment. Examples are available here (Zaitoun, Sagi, and Peleg 2024).

Results

Table 2 provides a comparative analysis of the overall paraphrasing performance of two models, GPT-4o and Llama 3, across the four ontologies: Pizza, People, Mondo, and FMA. The metrics displayed include true positives (TP), false positives (FP), and false negatives (FN). Additionally, the table presents precision, recall, and F1 scores for each model and ontology. GPT-4o consistently demonstrates high precision, recall, and F1 scores across all ontologies, indicating its robustness and reliability. In contrast, although Llama 3 also performs well, it shows greater variability in its precision and recall, especially in more complex ontologies such as Mondo and FMA. Overall, GPT-4o exhibits superior performance with higher precision, recall, and F1 scores compared to Llama 3.

Complex concepts are expected to have more CNL statements, leading to lower model performance. Therefore, we analyzed the performance of both models using a scatter plot and graphed the trend lines of precision, recall, and F1 (Figure 3). As expected, in both models, the more statements there are, the lower the recall. Conversely, for GPT-4o, the precision remained consistent regardless of the number of statements, whereas for Llama 3, the precision increased. Another major difference between the models is how recall decreases with increased statements. It has been observed that beyond 30 statements, the recall of Llama 3 drops significantly, while GPT-4o maintains a higher average. We performed statistical analysis to validate these observations and found that there is a significant (p-value < 0.05) negative correlation between the number of statements and the recall. However, neither a positive nor a negative statistically significant correlation was found for either LLMs regarding the precision. Finally, we wanted to see the impact of different OWL axioms on recall and precision. Thus, we performed the same correlation analysis on each axiom type individually and found the following:

1. Both *subClassOf* and *someValuesFrom* negatively impact the recall of both models.
2. For *GPT-4o*, the recall is also negatively impacted by *disjointWith*, *allValuesFrom*, and *unionOf*, but this impact is negligible.

Ontology	Classes	CNL State-ments	sub Class Of	some Values From	equiv. Class	∩	disjoint w.	all Values From	union Of	Other
Pizza	97	345	254	155	14	14	796	25	25	7
People	18	36	4	3	10	9	1	-	-	14
Mondo	100	437	196	57	8	8	-	-	-	-
FMA	107	723	487	413	7	30	32	-	-	-

Table 1: Evaluated Ontologies. The 8 columns show types of OWL axioms. The subClassOf is the is-a relationship. ∩ represents intersection.

Ontology	**Model**	**TP**	**FP (Extra)**	**FN (Misses)**	**Precision**	**Recall**	**F1**
Pizza	GPT-4o	331	1	14	0.997	0.959	0.978
	Llama 3	296	17	49	0.946	0.858	0.900
People	GPT-4o	36	0	0	1.000	1.000	1.000
	Llama 3	34	6	2	0.850	0.944	0.895
Mondo	GPT-4o	411	6	26	0.986	0.941	0.963
	Llama 3	325	33	112	0.908	0.744	0.818
FMA	GPT-4o	688	6	9	0.991	0.987	0.989
	Llama 3	475	27	247	0.946	0.658	0.776
Overall	GPT-4o	1466	13	49	0.991	0.968	0.979
	Llama 3 8B	1130	83	410	0.932	0.734	0.821

Table 2: GPT-4o and Llama 3 (8B) performance per ontology. Overall scores are computed using micro-averaging.

3. For *Llama 3*, the precision *increases* for axioms such as *disjointWith* and *allValuesFrom*, but *decreases* for *hasValue*.

Interestingly, for Llama 3, in some cases, the precision increases with the number of statements. This could be because the model is trained on tasks requiring generating a certain number of tokens for a given prompt. Therefore, when the input and the output are short, it attempts to fill this gap with additional information, resulting in false positives. When the statements are lengthy, it does not need to do so, and thus it does not generate additional tokens.

Importance of CNL Generation

We compare the verbalization output of the same LLMs directly on the OWL statements in the RDF Turtle format.. It can be observed that the LLMs have great success in paraphrasing the information available in OWL and depicting it as text. Yet, it comes off as robotic and unnatural because most of the technical jargon of OWL is carried along with it. Although they are correct, such samples would not be helpful as a dataset to train a model for the ontology learning task, as they do not represent the natural language present in the scientific documents used in this task well.

Discussion

Although the performance of the models reached very high precision and recall, we observed some recurring errors. Consequently, we conducted a qualitative analysis to classify the different types of errors and their corresponding OWL axioms. This helped us determine whether certain mistakes were associated with specific types of axioms or were recurring so that we could identify ways to improve the verbalizer. For GPT-4o, issues were identified with fewer than 50 classes across all four ontologies, so all were analyzed and classified. For Llama 3, 123 errors were analyzed for three ontologies. It is estimated to have more than 150 errors, but we only analyzed the results from the first three ontologies (Pizza, People, and Mondo).

In our analysis, we classified ten types of different errors, namely

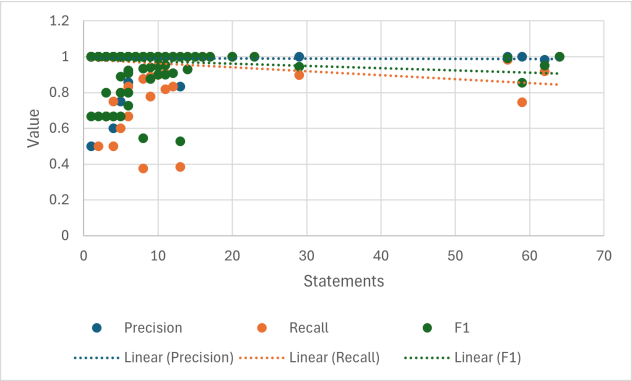
i. **Concept renamed** - the name of one of the concepts in a relationship (triple) was either changed or partially rephrased. For example, *cavity of humerus* is rephrased as *cavity found in the humerus*.

ii. **Missing relationship** - A relationship or triple that was completely omitted.

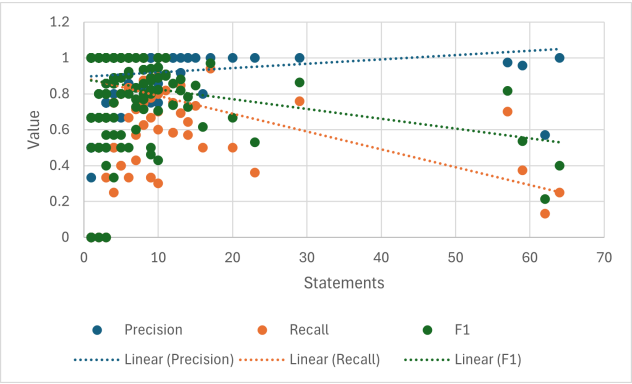
iii. **Relationship renamed** - the relationship was replaced with another relationship that has the same meaning. For example, *constitutional part* rephrased to *essential part*, or *Nerve1 nerve_supply_of Organ1* was replaced by the LLM as *Nerve1 supplies Organ1*.

iv. **Incorrect relationship** - the relationship was replaced with one of a different meaning. For example - replacing *part* (intending has part) with *part of*.

v. **Relationship with missing value** - the relationship was present, but not its value. For instance, the original CNL statement was *hot spiced beef topping is a type of at*



(a) Performance Analysis for GPT-4o



(b) Performance Analysis for Llama 3 (8B)

Figure 3: Scatter plots analyzing the performance of the two models based on the number of statements along with trend lines.

least has spiciness some a hot, but was paraphrased as *Hot spiced beef topping is a type of meat topping that has spiciness [hot was omitted].*

vi. **Incomplete decomposition** - Restructuring information in a way that is incomplete. Consider the following CNL statement from FMA: *posteromedial part of right side of middle part of peripheral zone of prostate* ***is regional part*** *of right side of middle part of peripheral zone of* ***prostate***. In the paraphrased version, the word *prostate* was completely omitted from the end of the sentence: *The posteromedial part of the right side of the middle part of the peripheral zone of the prostate* ***is a regional part of*** *the right side of the middle part of the peripheral zone* [*of the prostate* is missing].

vii. **Incorrect decomposition** - Restructuring information in a way that is incorrect. The following example demonstrates this type of error - *benign colon neoplasm is a type of a benign neoplasm of large intestine. benign colon neoplasm is a type of a colonic neoplasm. benign colon neoplasm is same as all of (a benign neoplasm, and at least disease has location some a colon)* - for this set of CNL statements, the model outputted the following paraphrased text: *It can be classified as a type of benign neoplasm affecting the large intestine, which includes the colon*. This example shows that the model broke apart three different statements (decomposed) and reconstructed something new incorrectly.

viii. **Model Laziness** - the paraphrased text contains partial information that is a part of a more extensive list. It is often observed when long lists are provided or a set of phrases that are repeated differently. In rare cases, the model would output phrases such as *"And others," "and many more," "etc."*. In other cases, it would simply omit certain elements of the original list.

ix. **Newly added information** - Adding or introducing additional information that is not present in the source. This type of error is easily identifiable and it refers to the majority of *False Positives*. Examples can be seen in document ((Zaitoun, Sagi, and Peleg 2024)), highlighted in blue.

x. **Incorrect paraphrasing** - this error refers to the misunderstanding of the original intent behind the source text and paraphrasing it as something different. This issue usually occurs with short CNL statements that do not have any additional context, making it difficult for the model to paraphrase. For example, the CNL statement *nitrogen compound transport is a type of a transport* was paraphrased as *Nitrogen compounds are transported through some kind of transportation method*, when the true intent behind the CNL was *Nitrogen compound transport is a type of transport*.

Axiom	GPT-4o	Llama 3 (8B)
subClassOf	46%	37%
allValuesFrom	6%	6%
someValuesFrom	26%	17%
disjointWith	4%	9%
equivalentClass	12%	7%
Not axiom-specific	6%	22%

Table 3: Error rate per axiom type for each model

We observed that the distribution of errors across different axioms is similar for both models (Table 3). The majority of errors occurred when a CNL statement was associated with the *subClassOf* axiom, accounting for 46% and 37% of errors for GPT-4o and Llama 3, respectively. These errors primarily happened when the models rephrased the original concept label or altered the relationship. The *someValuesFrom* axiom had the second highest error rate, with 26% of errors for GPT-4o and 17% for Llama 3, primarily due to missing or incorrect relationships. The full breakdown is available here (Zaitoun, Sagi, and Peleg 2024).

Since most issues arose from rephrasing classes or relationships, leading to some loss of information, we realized that these errors could be addressed with minimal effort through prompt engineering. By quoting each label and instructing the LLM to treat the label as a whole, we can prevent incorrect decomposition and preserve the original intent and information with high precision. However, this introduces a trade-off, which we refer to as the precision-vs-

fluency trade-off. While it is true that rephrasing classes or relationships result in a lossy conversion - meaning a model trained on ontology generation from text will not produce a precise set of OWL statements as the original - this change introduces more variability in the dataset. This variability could potentially result in a more fluent and natural dataset. This hypothesis will be addressed in future work.

We note that not all of the errors of the model were bad. In most cases, most of the false positives made by the models were additional information beyond what was in the source CNL statements; inspecting these additions, we observed that the information was correct and contributed to the newly paraphrased text.

Finally, the models demonstrated great composition abilities, restructuring statements to form new ones while retaining the original meaning. Such a case can be seen in the partial example of *Left Hip* from *FMA*.

```
1  left hip is a type of at least regional part of
       some a pelvic girdle region of left hip.
2  left hip is a type of must laterality a Left.
```

Which was paraphrased as:

```
1  The left hip is also part of the pelvic girdle
       region and the femoral region of the left hip.
```

Limitations and Future Work

While our work provides valuable insights into the performance and error distribution of our proposed LLM-assisted verbalizer, it is important to acknowledge several limitations that may impact the generalizability and comprehensiveness of our findings. First, our evaluation was conducted on a relatively small sample set. This limited scope may not fully capture the breadth of potential errors and variations in different contexts, potentially skewing the observed error rates and types. Second, our study utilized only two real-world ontologies. Although these ontologies provided a basis for testing, the results may not be representative of the performance across a wider range of ontologies with varying structures, complexities, and axioms. In future work, we plan to expand our evaluation to include a more diverse set of complex, real-world ontologies to better assess the robustness and applicability of our approach across different domains. Third, the evaluation was conducted using only two models, GPT-4o and Llama 3 (8B). While these models are advanced and widely used, including additional models of different sizes, as well as pre-trained on domain-specific data, could provide a more comprehensive understanding of the strengths and weaknesses of different models in the context of ontology verbalization. Furthermore, the issue of randomness in LLM outputs and the reproducibility of results remain significant challenges. Future work will focus on developing strategies to minimize randomness by carefully tuning model hyperparameters and conducting multiple iterations of experiments to observe how metrics change across different runs, providing a more robust analysis of the models' performance.

Conclusion

In this paper, we demonstrated our approach to verbalizing ontologies into natural language using LLMs. With GPT-4o, we achieved an overall precision of 0.99 and a recall of 0.96, while Llama 3 achieved a precision of 0.93 and a recall of 0.73. Despite the significant size difference between GPT-4o and Llama 3, the performance gap is not substantial. We anticipate that a slightly larger version of Llama 3 could match GPT-4o's performance, indicating that effective verbalization doesn't necessarily require very large models.

In our evaluation process, we evaluated the paraphrasing capabilities of LLMs with respect to the generated CNL statements, which are a direct representation of the OWL statements. We rewarded the model with a TP score when it accurately paraphrased a statement and punished it when a statement was missed with a score of FN or an additional statement with a score of FP. In many cases, the model added additional information that was not present in the original statements (FP) that were correct, yet because our evaluation metric is considering the task of paraphrasing, we punished the model regardless. In one of the examples, GPT-4o accurately predicted the acronyms for *triiodothyronine* and *reverse triiodothyronine* being *T3* and *rT3* respectively, despite never providing them within the original input.

While our study primarily focused on the use case of training data generation, our approach has the potential for much broader applications. In future work, we plan to explore other potential use cases, such as enhancing explainability and verbalizing general-purpose linked data, to extend the utility of our verbalizer beyond just training data generation.

Although we evaluated the performance of the models, a more interesting evaluation would be that of the model's performance trained on the produced datasets. We leave this effort for future work.

Our source code and datasets are available online[1].

Acknowledgments

This work was partially supported by the Data Science Research Center at the University of Haifa through the Israel PBC grant *Advancing Data Science to Serve Humanity and Protect the Global Environment* (Grant no. 100009443).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alobaidi, M.; Malik, K. M.; and Sabra, S. 2018. Linked open data-based framework for automatic biomedical ontology generation. *BMC bioinformatics*, 19(1): 1–13.

Androutsopoulos, I.; Lampouras, G.; and Galanis, D. 2013. Generating natural language descriptions from OWL ontologies: the NaturalOWL system. *Journal of Artificial Intelligence Research*, 48: 671–715.

[1]https://github.com/Minitour/ontology-verbalizer

Asim, M. N.; Wasim, M.; Khan, M. U. G.; Mahmood, W.; and Abbasi, H. M. 2018. A survey of ontology learning techniques and applications. *Database: The Journal of Biological Databases and Curation*, 2018.

del Valle, E. P. G.; García, G. L.; Ruiz, E. M.; Santamaría, L. P.; Zanin, M.; and Rodriguez-Gonzalez, A. 2019. Completing missing MeSH code mappings in UMLS through alternative expert-curated sources. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, 174–179. IEEE.

Dunn, A.; Dagdelen, J.; Walker, N. T.; Lee, S.; Rosen, A. S.; Ceder, G.; Persson, K. A.; and Jain, A. 2022. Structured information extraction from complex scientific text with fine-tuned large language models. *ArXiv*, abs/2212.05238.

Fuchs, N. E.; Kaljurand, K.; and Kuhn, T. 2008. Attempto controlled english for knowledge representation. *Reasoning Web: 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures*, 104–124.

Giglou, H. B.; D'Souza, J.; and Auer, S. 2023. LLMs4OL: Large Language Models for Ontology Learning. *ArXiv*, abs/2307.16648.

Gruber, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6): 907–928.

He, Y.; Chen, J.; Jimenez-Ruiz, E.; Dong, H.; and Horrocks, I. 2023. Language Model Analysis for Ontology Subsumption Inference. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 3439–3453. Toronto, Canada: Association for Computational Linguistics.

Horridge, M.; Drummond, N.; Goodwin, J.; Rector, A. L.; Stevens, R.; and Wang, H. 2006. The Manchester OWL syntax. In *OWLed*, volume 216.

Horridge, M.; Knublauch, H.; Rector, A.; Stevens, R.; and Wroe, C. 2004. A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools edition 1.0. *University of Manchester*.

Kaljurand, K. 2007. *Attempto controlled english as a semantic web language*. University of Tartu.

Kuhn, T. 2014. A survey and classification of controlled natural languages. *Computational linguistics*, 40(1): 121–170.

Marvin, G.; Hellen, N.; Jjingo, D.; and Nakatumba-Nabende, J. 2023. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, 387–402. Springer.

Meta. 2024. The LLaMA 3 Herd of Models.

Mille, S.; Dasiopoulou, S.; and Wanner, L. 2019. A portable grammar-based NLG system for verbalization of structured data. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 1054–1056.

Mountasser, I.; Ouhbi, B.; Hdioud, F.; and Frikh, B. 2021. Semantic-based Big Data integration framework using scalable distributed ontology matching strategy. *Distributed and Parallel Databases*, 39: 891–937.

Peleg, M.; Veggiotti, N.; Sacchi, L.; and Wilk, S. 2024. How can we reward you? A compliance and reward ontology (CaRO) for eliciting quantitative reward rules for engagement in mHealth app and healthy behaviors. *Journal of Biomedical Informatics*, 154: 104655.

Rosse, C.; and Mejino Jr, J. L. 2003. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of biomedical informatics*, 36(6): 478–500.

Taglino, F.; Cumbo, F.; Antognoli, G.; Arisi, I.; D'Onofrio, M.; Perazzoni, F.; Voyat, R.; Fiscon, G.; Conte, F.; Canevelli, M.; Bruno, G.; Mecocci, P.; and Bertolazzi, P. 2023. An ontology-based approach for modelling and querying Alzheimer's disease data. *BMC Medical Informatics and Decision Making*, 23.

Vasilevsky, N. A.; Matentzoglu, N. A.; Toro, S.; Flack IV, J. E.; Hegde, H.; Unni, D. R.; Alyea, G. F.; Amberger, J. S.; Babb, L.; Balhoff, J. P.; et al. 2022. Mondo: Unifying diseases for the world, by the world. *MedRxiv*, 2022–04.

Vidanage, K.; Mohemad, R.; Noor, N. M. M.; and Bakar, Z. A. 2021. Fully Automated Ontology IncrementS user Guide Generation. *International Journal of Advanced Computer Science and Applications*, 12(6).

Wallace, R. S. 2009. *The anatomy of ALICE*. Springer.

Watrobski, J. 2020. Ontology learning methods from text - an extensive knowledge-based approach. In *International Conference on Knowledge-Based Intelligent Information & Engineering Systems*.

Web Ontology Language. 2012. OWL 2 Web Ontology Language Document Overview (Second Edition). W3c recommendation, W3C. Accessed on July 19, 2024.

Yang, L.; Cormican, K.; and Yu, M. 2019. Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry*, 111: 148–171.

Zada, I.; Shahzad, S.; Ali, S.; and Mehmood, R. M. 2023. OntoSuSD: Software engineering approaches integration ontology for sustainable software development. *Software: Practice and Experience*, 53(2): 283–317.

Zaitoun. 2024a. Text to OWL with Prompt Engineering. https://github.com/Minitour/ontology-verbalizer/wiki/NL-to-OWL-using-Prompt-Engineering.

Zaitoun. 2024b. Verbalizer Algorithm. https://github.com/Minitour/ontology-verbalizer/wiki/CNL-Generation-Algorithm.

Zaitoun, A.; Sagi, T.; and Peleg, M. 2024. Generating Ontology-Learning Training-Data through Verbalization (Appendix). *https://doi.org/10.5281/zenodo.13776828*.

Zaitoun, A.; Sagi, T.; Wilk, S.; and Peleg, M. 2023. Can Large Language Models Augment a Biomedical Ontology with missing Concepts and Relations? *arXiv preprint arXiv:2311.06858*.