



Do LLMs Really Adapt to Domains? An Ontology Learning Perspective

Huu Tan Mai^{1,2} , Cuong Xuan Chu² , and Heiko Paulheim¹ 

¹ University of Mannheim, Data and Web Science Group, 68131 Mannheim, Germany
{[hhuu.tan.mai](mailto:hhuu.tan.mai@uni-mannheim.de), [heiko.paulheim](mailto:heiko.paulheim@uni-mannheim.de)}@uni-mannheim.de

² Bosch Center for AI, 71272 Renningen, Germany
{[huutan.mai](mailto:huutan.mai@de.bosch.com), [cuongxuan.chu](mailto:cuongxuan.chu@de.bosch.com)}@de.bosch.com

Abstract. Large Language Models (LLMs) have demonstrated unprecedented prowess across various natural language processing tasks in various application domains. Recent studies show that LLMs can be leveraged to perform lexical semantic tasks, such as Knowledge Base Completion (KBC) or Ontology Learning (OL). However, it has not effectively been verified whether their success is due to their ability to reason over unstructured or semi-structured data, or their effective learning of linguistic patterns and senses alone. This unresolved question is particularly crucial when dealing with domain-specific data, where the lexical senses and their meaning can completely differ from what a LLM has learned during its training stage. This paper investigates the following question: *Do LLMs really adapt to domains and remain consistent in the extraction of structured knowledge, or do they only learn lexical senses instead of reasoning?* To answer this question and, we devise a controlled experiment setup that uses WordNet to synthesize parallel corpora, with English and gibberish terms. We examine the differences in the outputs of LLMs for each corpus in two OL tasks: relation extraction and taxonomy discovery. Empirical results show that, while adapting to the gibberish corpora, off-the-shelf LLMs do not consistently reason over semantic relationships between concepts, and instead leverage senses and their frame. However, fine-tuning improves the performance of LLMs on lexical semantic tasks even when the domain-specific terms are arbitrary and unseen during pre-training, hinting at the applicability of pre-trained LLMs for OL.

Keywords: ontology learning · LLMs · domain adaptation

1 Introduction

Knowledge Bases (KB) and ontologies play a key role in structuring and organizing knowledge across domains, and offer powerful solutions to link data that would otherwise remain unstructured (such as text). As of now, many sources of data can be used as ontologies, of varying specificity. For instance, WordNet [17], ConceptNet [14] and WebIsA [23] contain common knowledge, whereas KBs such

as Unified Medical Language System (UMLS) [2] and GeoNames [6] have their own domain specificities (respectively, medical and geographic). For knowledge-intensive applications, access to structured data is of utmost importance, but creating it is a very tedious and time-consuming process that inevitably demands domain expertise. Ontology Learning (OL) is a field of artificial intelligence that concerns itself with automatically identifying terms, types and axioms between them from unstructured or structured information sources such as text or KBs. In particular, OL deals with identifying hyponymy (resp. hypernymy) relations in a KB. That is, for pairs of concepts (A, B) in the KB, one wants to infer whether or not Concept A is a subclass (resp. a superclass) of Concept B .

In the past few years, Large Language Models (LLMs), such as GPT-3 [3], GPT-4, LLaMa2 [26] or Falcon-40B [1] have displayed unprecedented prowess in many NLP applications across various domains. LLMs are language models with a very large parameter count that are trained with enormous amounts of textual data. Hence, they are equipped with common knowledge and have shown remarkable success in generating text. Furthermore, LLMs have made it possible to capture the meaning of text and reason about it, providing a promising alternative for knowledge-intensive tasks such as KB completion and ontology-related tasks like OL and OM (Ontology Matching). Recent studies have shown that LLMs could be viewed as Knowledge Bases [22], storing knowledge incorporated in their parameters (e.g. factual knowledge [25], event knowledge [11], commonsense knowledge [13, 33], etc.). In particular, LLMs4OL [7] introduces a novel approach to OL using LLMs: the work provides comprehensive empirical evidence that, although requiring fine-tuning for better performance, LLMs can work as effective ontology learners on specialized datasets.

However, there are several challenges that come with LLMs, and that have been left relatively unexplored in the frame of ontology learning. They include: (1) LLMs are susceptible to hallucinate [12], i.e. provide generate text that is syntactically sound but factually incorrect. (2) LLMs are trained on massive corpora of textual data and acquire common knowledge, but their few - or zero - shot generalizability and adaptability to unknown domains remains relatively undiscussed. Studying these two aspects is crucial to better understand the current limits of LLMs, and is ever so needed since they are being become increasingly prominent for domain-specific uses. On the one hand, it is effectively possible to get LLMs to *adapt* to a domain by fine-tuning, but such a process requires labeled data in this domain and can be computationally expensive. On the other hand, *generalizability* is also an extremely valuable quality for OL in domain-specific settings. To illustrate this, consider the Examples 1 and 2.

In OL, a LLM may be able to identify that a *macaron* is a subclass of *confection* that is made from many ingredients such as *egg white*, *icing sugar*, *granulated sugar*, and so on. Given Example 2, obtained by turning the words of Example 1 into gibberish, a model capable of *generalization* should retrieve the analogous concepts as hypernyms or meronyms (i.e., *twiglomptoroa* is a subclass of *becsverdecoroal* etc.). However, it was not explicitly verified if LLMs would do so, or more broadly, if they are able to generalize taxonomic axioms over text

Example 1 (macaron). a sweet meringue-based confection made with egg white, icing sugar, granulated sugar, almond powder or ground almond, and food colouring.

Example 2 (twiglomptroa). a becsverdecoroa meringue-based becsverdecoroal made with egg white, icing shifbousadu stintsobroa, granulated shifbousadu stintsobroa, almond powder or ground almond, and food colouring.

(e.g. textual patterns indicating that A is a subclass of B) rather than learn the concepts themselves during the pre-training process.

This paper presents comprehensive experiments to study the generalizability and domain adaptability of LLMs, with perspective of ontology learning. By synthesizing three new domain corpora from the Open English WordNet, and creating a gibberish counterpart each, we assess the performance of LLMs on two main tasks: relation extraction and taxonomy discovery. We conduct the experiments on popular LLMs, including closed- and open-source, off-the-shelf and fine-tuned, and evaluate them on both in-domain and across-domain setups. The novel contributions of this paper are as follows:

- We create three synthetic datasets as parallel corpora from the Open English WordNet, by turning domain-specific terms into gibberish.
- We conduct experiments to simulate the adaptability and generalizability of LLMs in unseen domains, with or without backpropagation.
- We provide empirical evidence that there is a limit to the generalization capability of off-the-shelf LLMs, with OL tasks that leverage lexical semantics such as relation extraction and taxonomy discovery.
- We show that in-domain fine-tuning improves in-domain task-specific performance, and that the improvements are transferable to new domains.

2 Related Work

Ontology Learning with LLMs. OL is the (semi)automatic acquisition of T-Box and/or A-Box data from various data sources. In the context of this work, we look at OL from unstructured text or semi-structured data such as Knowledge Graphs paired with textual descriptions. More generally, recent studies show that LLMs can be leveraged to perform ontology related tasks, such as OM (Ontology Matching) or OL. For instance, Norouzi et al. [20] use a naive approach that uses ChatGPT for ontology alignment, by providing the entire source and target ontologies. Mateiu et al. [15] use a fine-tuned GPT-3 to convert natural language into OWL Functional Syntax for ontology enrichment. Hertling et al. [9] use few-shot prompting to enhance the performance of open-source LLMs on OM tasks.

In LLMs4OL [7], the authors argue that with sufficient formulation, all tasks pertinent to OL fall within one of the three categories: A) Term Typing (determining a generalized type for a lexical term), B) Taxonomy Discovery (determining the hierarchy between a pair of concepts), C) Non-Taxonomic

Relation Extraction (finding non-hierarchic relations between concepts). This task paradigm allows them to evaluate LLMs in OL with a zero-shot prompting method. In the work, the authors show that although LLMs may not be suitable for OL as is, they may still be helpful when effectively fine-tuned for ontology construction.

SPIRES [4] is a successful application of LLMs to populate ontologies. It leverages Zero-Shot Learning to extract relations between concepts in textual corpora, then grounds the concepts using other existing ontologies in the target domain (e.g. FoodOn or Wikidata). However, in a domain-specific setting, it is not granted that public ontologies of the domain exist and are of high quality.

Moskvoretskii et al. [19] use LLaMa [26] fine-tuned on WordNet, to perform OL tasks such as taxonomy discovery, taxonomy enrichment, taxonomy construction and lexical entailment. Specifically, they provide further empirical evidence that fine-tuning LLMs for taxonomy discovery on domains drastically increases their performance, making them suitable for the task. The method was tested on real domains such as the food, music and medical domains. In comparison, our work seeks to establish whether or not domain adaptation would hold in arbitrary domains, where the terminology is unknown to the model.

In-Context Learning with LLMs. It was previously observed and verified that LLMs can learn from a few in-context examples in the form of demonstration. In fact, to better answer a given query, a LLM can leverage previous examples to estimate the distribution of input-output pairs. This emergent behavior of LLMs [29] has become a successful learning paradigm because it no longer requires the expensive optimization of model parameters. With respect to our paper, four particular works pertaining to In-Context Learning (ICL) are of high interest. Firstly, Chain-of-Thought prompting [30] forces the model to generate intermediate steps before returning an output, which was shown to elicit reasoning in very large language models and improve their symbolic reasoning performance. Secondly, Min et al. [18] show that ground-truth labels do not significantly hurt the performance of LLMs on downstream tasks, suggesting that models implicitly learn input-label mappings from the language modelling objective alone. Thirdly, symbol tuning [31], the process of fine-tuning by replacing original labels to semantically unrelated ones, was shown to improve the in-context learning performance of very large LMs, and effectively override prior semantic knowledge. Finally, Wei et al. [32] show that “smaller” LLMs greatly suffer from semantically unrelated labels in comparison to larger ones, heavily implying that they overly rely on semantic priors of targets instead of effectively reasoning over them. The contributions of these works justify our necessity to verify the in-context adaptation of LLMs to domains by extending this verification to semantically unrelated inputs (e.g. gibberish input-label mappings).

Domain Adaptation with LLMs. Few works of interest deal with the performance of Large Language Models given domain-specific training corpora. Wan et al. [28] propose a domain adaptation framework for very large language models (GPT-4 [21]) to address the scarcity of Chinese legal domain texts, using an adapt-retrieve-revise process. With a smaller LLM trained on in-domain cor-

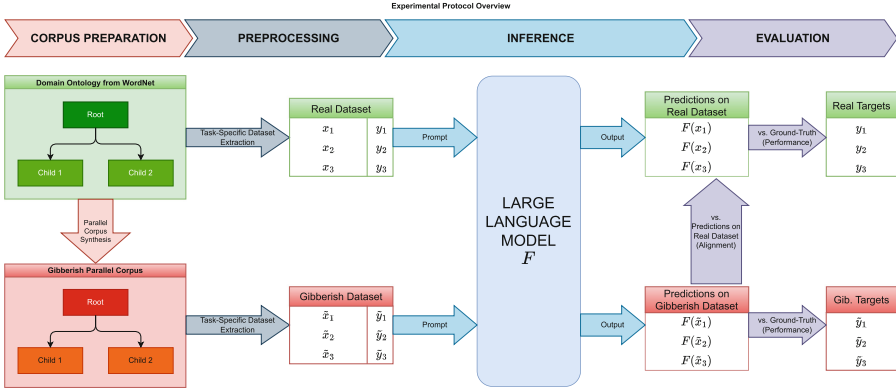


Fig. 1. Overview of the off-the-shelf evaluation pipeline. The dataset extraction and the prompt depend on the task to perform.

pora, the authors generate a draft answer to retrieve evidence from an external knowledge base, both of which are given to GPT-4 to generate a final answer. Gururangan et al. [8] show that domain-adaptive pretraining (DAPT), i.e. continued pre-training on domain-specific text, allows one to adapt a language model to a domain at reduced cost. However, Cheng et al. [5] argue that while DAPT may improve specific task performance after fine-tuning, it overall hurts the ability of LLMs to perform question answering. Instead, they propose AdaptLLM, a scalable approach that aims to train a LLM on reading comprehension texts created from the raw domain-specific corpora. Finally, Shin et al. [24] show that the in-context learning ability of a LLM heavily depends on the corpus sources, and may emerge by combining multiple corpora, but the domain relevance of the corpus may not be indicative of the few-shot performance of the model.

3 Approach

Figure 1 illustrates an overview of the pipeline we employ to test the *adaptability* and *generalizability* of off-the-shelf LLMs (pretrained, and as is), which includes two main steps: corpus preparation and LLMs evaluation.

In particular, we use the English WordNet (2023 Edition) [16] to generate a parallel corpus of terms and definitions in the form of gibberish, which serve as our reference domain-specific setting. The process begins by choosing root concepts (for instance, sweets and desserts) in the WordNet taxonomy, then explores the graph through hyponymy, derivation and other (e.g. topic) relationships across concepts (with a set maximal exploration depth). The explored concepts form a domain in the real WordNet (e.g. sweets domain) that can be used to create a parallel corpus by propagating gibberish representations and definitions. More details about the algorithm can be found in Sect. 3.1.

After obtaining a parallel corpus of a particular domain, we evaluate a LLM on two different tasks: *relation extraction* and *taxonomy discovery*, each on both versions of the corpus. Naturally, since the concepts remain the same up to an input-label mapping, we ideally expect the results to be analogous. More details can be found in Subsect. 3.2. Additionally, we investigate the effect of fine-tuning on the in-domain performance of LLMs, as described in Subsect. 3.3.

3.1 Parallel Corpus Synthesis

To simulate a domain that is unseen for the LLM, we generate another KG where the domain concepts have gibberish representations and definitions that do not collide (e.g. if “sugar” is turned into “arghl” then any definition that contains the word “sugar” will see it turned into the word “arghl” instead). For this purpose, we devise a procedure which includes three steps: concept mining, concept linking and gibberish generation. The code can be found online.¹

Concept Mining. The concept mining algorithm is a simple Breadth-First Search algorithm, starting from each of the root concepts C_0 and only going through user-selected relationships (hypernymy, sense derivation, and concept topic). We set a maximal exploration depth D , which is set to 5 during experiments. The explored concepts form a dataset \mathcal{D} .

Concept Linking. The next step of this process is to establish the dependencies between concept definitions and other concepts. For example, if “sugar” is mentioned in the definition of concept c , then we will link all the concepts c_{sugar} which have “sugar” as a representation with c by introducing a blank node as follows, where $\{c_id\}$ denotes the WordNet ID of c :

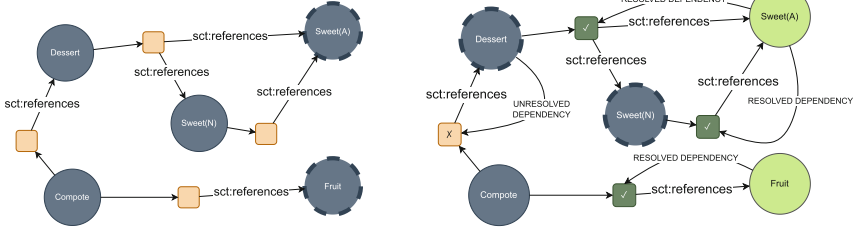
```
1 c      sct:definitionWord      _:_sugar_{c_id} .
2 _:_sugar_{c_id} rdf:value      "sugar"@en .
3 _:_sugar_{c_id} sct:references  c_sugar .
```

Listing 1.1. Example of linking triples.

Gibberish Generation. We assume that we have an algorithm T that creates a gibberish representation $T(c)$ from a concept c based on its initial representation, definition and part-of-speech. A concept is fully processed when it has a gibberish definition AND a gibberish representation. A concept is partially processed when it has a gibberish representation (fully processed implies partially processed).

Denote \mathcal{D}_0 the set of concepts in \mathcal{D} that have no internal dependencies (i.e. for any c in \mathcal{D}_0 , the definition of c does not refer to any concept in \mathcal{D}), as shown in Fig. 2a. We create an initial representation $T(c)$ for any c in \mathcal{D}_0 , and give them a gibberish definition identical to their original one. We will additionally

¹ <https://github.com/boschresearch/llm-vs-gibberish-ontologies>.



(a) Step 0 : Make a gibberish representation and (unchanged) definition for concepts with no dependencies.

(b) Step $n+1$: Use at least partially processed concepts to resolve dependencies and create new gibberish representations and definitions.

Fig. 2. Algorithm to propagate gibberish representations and definitions. Squares are blank nodes that indicate that the preceding concept has a word that refers to one of the node’s successors. Dashed circles represent the concepts eligible to be fully processed. Green circles represent (at least partially) processed concepts.

add the homonyms of the previously processed concepts in \mathcal{D}_0 , with no gibberish representation. Moreover, set $\mathcal{D}_{-1} = \emptyset$. We then repeat the following loop until we have fully processed all the concepts in \mathcal{D} , as illustrated in Fig. 2b.

Suppose we have obtained \mathcal{D}_{n-1} and \mathcal{D}_{n-2} . If $|\mathcal{D}_{n-1}| > |\mathcal{D}_{n-2}|$, we proceed as follows: obtain concepts c that are not fully labeled and which dependencies can all be resolved (i.e. for each dependency c `sct:definitionWord` x , there exists $c' \in \mathcal{D}_{n-1}$, or $c' = c$ such that x `sct:references` c'). We create gibberish representations $T(c)$ if there is not one already, we resolve the dependencies using the gibberish representations in \mathcal{D}_{n-1} to make a definition for c . All homonyms of processed concepts are also partially processed, and all the partially processed concepts are added to \mathcal{D}_{n-1} to form \mathcal{D}_n .

If $|\mathcal{D}_{n-1}| = |\mathcal{D}_{n-2}|$, we sample a random concept c of \mathcal{D} that is not partially processed, assign a gibberish representation $T(c)$ to it and add c to \mathcal{D}_{n-1} to form \mathcal{D}_n . Note that for $i \geq 0$, \mathcal{D}_i does not exclusively contain concepts in \mathcal{D} , but $\bar{\mathcal{D}}_i = \mathcal{D}_i \cap \mathcal{D}$ does.

This pipeline yields a set of concepts \mathcal{D} , where each concept $c \in \mathcal{D}$ has a gibberish representation $T(c)$ as well as a gibberish definition consistent with the internal dependencies in \mathcal{D} .

Example. Consider the example depicted in Fig. 2. In the first step, Sweet (adjective) and Fruit do not reference any other nodes in their definitions: they can be fully processed. In the second step, since Dessert refers to both Sweet (noun) and Sweet (adjective) and the latter was processed, it is eligible to be processed next. Moreover, since Sweet (adjective) is a homonym of Sweet (noun), it can also be processed. However, since Compote refers to both Dessert and Fruit, it may not be processed yet, since Dessert has not been processed yet. It will be processed in the following step.

3.2 Off-the-Shelf Evaluation Methodology

For each dataset \mathcal{D} , we perform an evaluation by comparing the performance of off-the-shelf LLMs in two different lexical semantic tasks: relation extraction and taxonomy discovery; and compare the outputs coming from two cases: on the original dataset, and its gibberish counterpart.

Relation Extraction. Given a query concept in \mathcal{D} , its lexical senses (i.e. written forms) and its definition, the goal of this task is to extract all relations between the concepts mentioned in the definition, including the query concept. To remain in the frame of ontology learning, we focus only on hypernymy and holonymy relationships. For instance, in Example 1, the extracted relations should be as follows: macaron is a subclass of confection, egg white is a part of macaron, icing sugar is a part of macaron, etc. Likewise, given the gibberish definition in 2 instead, we expect the same relations to be extracted, only with the concept names replaced with their gibberish counterpart. In order to retrieve a prediction, the LLM is prompted to output triples in the form: (A, r, B) where r is either `is a subclass of` or `is a part of`.

Taxonomy Discovery. Given two concepts A and B in \mathcal{D} , with their lexical senses and definitions, the goal of this task is to determine whether or not A is a subclass of B. Likewise, we expect that turning the lexical senses and definitions of A and B into gibberish will not change the outcome. WordNet hypernymy relations are used as a ground-truth: the predictions on the real (resp. gibberish counterpart of the) dataset are compared with the real (resp. gibberish counterpart of the) ground-truth. Indirect hypernymy relations (obtained with the transitive closure of the relation `gwn:hypernym`) are also used. Negative examples are produced by corrupting the hypernym once or twice per query (hyponym, subclass of, ?). This classification task can be evaluated with F1-score: a drop in performance should indicate that a LLM relies on the lexical senses to infer taxonomical relationships rather than textual semantic information.

Prompting. For each task/model configuration, we follow general guidelines for prompting a LLM. (1) The return is in JSON format, (2) Chain-of-Thought (CoT) [30] can be used if it improves the performance of the model, (3) One or few exemplar(s) can be used, with example(s) outside of the dataset, if it improves the performance of the model. In the **relation extraction** task, for a concept C , its written form F_C , its part-of-speech P_C , and its definition D_C , we construct a prompt $p(C)$ as shown in Listing 1.2. In the **taxonomy discovery** task, for A, B two concepts, F_A, F_B their written forms, D_A, D_B their definition, we construct a prompt $p(A, B)$ as shown in Listing 1.3. Prompt templates can be found [here](#).


```

1 {FORMAT INSTRUCTIONS (Task and return format)}
2 {EXAMPLE(S) (zero, one or few exemplars)}
3
4 Concept: {F_C}
5 Part-of-speech: {P_C}
6 Definition: {D_C}

```

Listing 1.2. Relation extraction generic prompt $p(C)$.

```

1 {FORMAT INSTRUCTIONS (Task and return format)}
2 {EXAMPLE(S) (zero, one or few exemplars)}
3
4 Concept A: {F_A}
5 Definition: {D_A}
6
7 Concept B: {F_B}
8 Definition: {D_B}

```

Listing 1.3. Taxonomy discovery generic prompt $p(A, B)$.

3.3 Fine-Tuning Experiment

While it was previously verified that fine-tuning improves OL performance in existing domains [19], it is not clear if this statement still holds for arbitrary domains and unknown vocabularies where reasoning is required. To verify this question, after evaluating the zero/one/few-shot performance of LLMs in the gibberish domains, we propose to assess the effect of fine-tuning models on the inference performance for a specific task.

Data Split. For each dataset \mathcal{D} , we fine-tune a LLM for taxonomy discovery on a train split of hypernymy relations. Half of the concepts in the dataset, and their hypernymy relations, are used for training. The inverse relations are used as negatives (if A is a subclass of B, then B is not a subclass of A), alongside some randomly sampled negatives. The remaining relations are used for testing.

Prompting. We train LLMs on instructive datasets with the prefix prompt in Listing 1.4, completed using each pair of concepts in the hypernymy dataset.

```

1 ### HUMAN:
2 Identify whether the statement is true or false. Answer
   with only one word: 'True' or 'False'.
3
4 CONCEPT A: {term_a} ({pos_a})
5 Definition: {definition_a}
6
7 CONCEPT B: {term_b} ({pos_b})
8 Definition: {definition_b}
9

```

Table 1. Dataset statistics. The ‘Hypernyms’ column indicates the number of retrieved hypernymy relationships within the dataset. The ‘Depth’ column indicates the exploration depth of the algorithm.

\mathcal{D}	Concepts	Hypernyms	Depth	Root Concepts
WN-sweets	244	418	5	sweet (n), sweet (a), sugar
WN-football	937	1401	5	football, team, offensive (a), defensive (a)
WN-music	1366	2497	5	musical instrument

```

10 Statement: '{term_a}' is a subclass of '{term_b}'.
11 ### ASSISTANT:

```

Listing 1.4. Prefix prompt used for the fine-tuning experiment. The content between curly brackets is formatted for each pair of concepts in the dataset.

4 Experiments

In this section, we explore the off-the-shelf evaluation proposed in Subsect. 3.2, and the fine-tuning experiments described in Subsect. 3.3.

4.1 Experimental Setup

Datasets. To assess the performance variation with domain specificity, we generate three synthetic domain-specific datasets as parallel corpora from the Open English WordNet [16], using the methodology described in Sect. 3.1. They are:

- Sweets: A collection of concepts related to sweets, desserts, sweet food or sugar. In this dataset, hypernyms are frequent, and concepts are relatively well constructed from their hypernyms.
- Football: A collection of concepts related to football. This dataset, created by browsing co-topic concepts, includes less taxonomic relationships, but has its own terminology and jargon.
- Music: A collection of concepts related to musical instruments. It is the largest of the three.

Table 1 shows, for each dataset, the number of concepts, the number of hypernymy relationships, the exploration depth and the root concepts. The translator T used to generate gibberish representations of concepts is available online.²

Models. While this study is not comprehensive of all existing LLMs, our goal is to show a particular trend between them. In the off-the-shelf evaluation, we evaluate the following popular LLMs, with their number of parameters between parentheses: GPT-3.5 [3] (174B), GPT-4 [21] (≥ 1 T), Falcon-40B [1] (40B),

² <https://github.com/htmai-880/gibberify>.

LLaMa2-13B [26] (13B), and Zephyr-7B- β [27] (7B). The former two, accessed with a paid subscription, are closed-source, whereas the latter three are open-source. In the fine-tuning experiment, we consider Zephyr-7B- β [27] and Falcon-7B [1], which are both open-source. For the paid subscription models, We limit our budget to 15€ per dataset.

Specificities. We henceforth consider the three following evaluation methods:

- Ground-truth (GT)(**en**) vs **en**: we compare the answers on the original English dataset against the ground-truth
- Ground-truth (GT)(**gib**) vs **gib**: we compare the answers on the gibberish dataset against the gibberish ground-truth
- **en** vs **gib**: using the answers on the original dataset as a ground-truth, we evaluate the consistency of the predictions on the gibberish dataset, regardless of their correctness.

In the relation extraction (GT(X) vs X), due to the scarcely annotated holonymy relationships in WordNet, we focus on hypernymy relationships to compute metrics. Moreover, a model prediction is processed by taking into account the inferred hypernymy relationships, using the transitive property of the relation. For instance, if a model outputs the triples (vanilla pudding, is a subclass of, custard-like pudding), and (custard-like pudding, is a subclass of, pudding), we count the triple (vanilla pudding, is a subclass of, pudding) as effectively predicted by the model. In the taxonomy discovery task, predictions that are neither “True” or “False” are ignored. Thus, macro-averaged F1-scores may not be between the macro-averaged precisions and the macro-averaged recalls.

4.2 Off-the-Shelf Evaluation

Metrics. For each model/task configuration, we compute metrics in three settings: GT(**en**) vs **en**, GT(**gib**) vs **gib**, and **en** vs **gib**. The goal is to see if the predictions with gibberish terms align with the predictions with the real terms. We compute the following metrics: precision, recall and F1-score.

Results. We first examine the results with respect the to ground-truths. It is important to mention that the English WordNet is scarcely annotated in terms of hypernymy and holonymy relationships. Consider the following example:

Example 3 (toffee apple). an apple that is covered with a candy-like substance (usually caramelized sugar).

In this example, the definition obviously implies that a *toffee apple* is an *apple*, but WordNet only considers *sweet*, *confection* to be valid hypernyms. The incompleteness of WordNet explains why the observed performances are so low, but because our goal is a relative comparison of performances on real corpora

Table 2. Results on the relation extraction task in the GT(X) vs X setting, where X is either **en** or **gib**. Only hypernymy relations are considered.

Model	X	WN-sweets			WN-football			WN-music		
		Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
GPT-3.5	en	.478	.150	.228	.383	.056	.097	.397	.060	.104
	gib	.336	.069	.115	.371	.035	.065	.307	.029	.053
GPT-4	en	.583	.160	.251	—	—	—	—	—	—
	gib	.530	.129	.207	—	—	—	—	—	—
Falcon-40B	en	.573	.151	.238	.489	.067	.118	.529	.065	.116
	gib	.330	.080	.128	.382	.050	.088	.341	.042	.074
LLaMa2-13B	en	.536	.141	.223	.423	.035	.065	.434	.030	.056
	gib	.365	.085	.138	.341	.018	.035	.296	.014	.026
Zephyr-7B- β	en	.441	.158	.233	.399	.067	.115	.374	.063	.108
	gib	.243	.095	.137	.313	.052	.089	.261	.044	.075

and their gibberish counterpart, rather than absolute scores, it is not critical to have a high-quality ground-truth.

In both tasks, which results are reported in Table 2 and Table 3, a common trend occurs across all LLMs and in all synthetic domains: a significant performance decrease is observed when replacing real terms with gibberish. Although GPT-4 (tested on WN-sweets only because of the slowness and the costs) performs best on gibberish corpora, it suffers from the same performance drop.

While the performance on relation extraction is generally low across all LLMs with the real datasets, which we mainly attribute to the poor quality of the annotations, it is even lower with the gibberish datasets. In each case, the macro F1-score is practically halved, e.g. in WN-sweets, GPT-3.5 drops from 0.228 to 0.115, Falcon-40B drops from 0.238 to 0.128, LLaMa2-13B drops from 0.223 to 0.138, Zephyr-7B- β drops from 0.233 to 0.137. Note that the recall is low in comparison to the precision, which indicates that a LLM tends to overlook indicators that two concepts are hierarchically related. This observation aligns with the conclusion of LLMs4OL [7], according to which off-the-shelf LLMs are not sufficiently suitable for OL tasks, particularly in the case of relation extraction.

In the taxonomy discovery task, the performance of LLMs also plummets when using the gibberish dataset instead of the real one. The LLMs are relatively good at identifying whether two real concepts are hierarchically related (e.g. F1-score up to 0.940 on the WN-sweets dataset by GPT-3.5), but suffer from a large performance drop when confronted to unknown words which share the same semantic relations (e.g. the F1-score of GPT-3.5 drops from 0.940 to 0.446 when using the gibberish counterpart of WN-sweets; the same behavior is observed across all datasets and LLMs). We interpret this drop as the fact that LLMs are significantly better at leveraging semantic priors (i.e. lexical senses known from pre-training), to deduce that Concept A is a subclass of Concept B.

Table 3. Results (macro-average) on the taxonomy discovery task in the GT(X) vs X setting, where X is either **en** or **gib**.

Model	X	WN-sweets			WN-football			WN-music		
		Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
GPT-3.5	en	.944	.937	.940	.758	.701	.648	.829	.858	.818
	gib	.783	.539	.446	.640	.505	.333	.687	.537	.361
GPT-4	en	.949	.943	.945	—	—	—	—	—	—
	gib	.591	.576	.566	—	—	—	—	—	—
Falcon-40B	en	.775	.658	.598	.800	.648	.637	.787	.620	.613
	gib	.591	.575	.574	.483	.475	.478	.541	.535	.480
LLaMa2-13B	en	.819	.772	.750	.808	.811	.809	.785	.800	.790
	gib	.450	.447	.444	.533	.531	.504	.576	.556	.465
Zephyr-7B- β	en	.899	.897	.898	.813	.751	.759	.821	.762	.778
	gib	.691	.634	.621	.500	.500	.469	.530	.524	.523

Table 4. Alignment results (macro-average) on the relation extraction task in the **en** vs **gib** setting. Both hypernymy and holonymy relations are considered.

Model	WN-sweets			WN-football			WN-music		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
GPT-3.5	.371	.304	.334	.263	.175	.210	.207	.138	.166
GPT-4	.504	.527	.515	—	—	—	—	—	—
Falcon-40B	.310	.303	.306	.236	.238	.237	.225	.224	.225
LLaMa2-13B	.347	.340	.344	.386	.225	.284	.374	.215	.273
Zephyr-7B- β	.214	.229	.221	.192	.180	.185	.148	.142	.145

Although the general performance drop is expected because the tested LLM would then be dealing with words it has never seen during its training, we furthermore observe that the prediction alignment is very low. In spite of analogous concepts sharing the same semantic relations with each other in the parallel corpora, the model is unable to produce analogous outputs from analogous inputs, as shown by the low F1-scores in Tables 4 and 5. This is evidence that as is, LLMs do not *reason* over semantic relationships.

Our interpretation is that the attention mechanism heavily relies on the lexical sense and the frame of a token, instead of leveraging the semantic relationships that hold between tokens. In other words, the “reasoning” abilities of LLMs for Ontology Learning are mostly limited to entities and concepts that the models have already been trained on, i.e. prior semantics. However, such a quality is critical for Ontology Learning in arbitrary domains, where hypernymy relationships must be retrieved for unknown concepts, or new concepts that share the same lexical form as some existing word (for instance, if domain-specific jargon employs existing words with different meanings).

Table 5. Alignment results (macro-average) on the taxonomy discovery task in the **en** vs **gib** setting.

Model	WN-sweets			WN-football			WN-music		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
GPT-3.5	.789	.541	.465	.700	.517	.488	.738	.544	.459
GPT-4	.611	.594	.590	—	—	—	—	—	—
Falcon-40B	.541	.557	.412	.502	.495	.443	.529	.562	.352
LLaMa2-13B	.565	.570	.556	.618	.610	.586	.641	.599	.535
Zephyr-7B- β	.818	.727	.716	.557	.544	.545	.570	.572	.570

Table 6. Hypernym train-test split for the fine-tuning experiment.

\mathcal{D}	Train		Test	
	Positives	Negatives	Positives	Negatives
WN-sweets	189	393	229	284
WN-football	674	1043	727	567
WN-music	1367	1882	1130	851

This trend seems to hold true in both tasks, which confirms the fact that off-the-shelf LLMs are currently not suited for OL tasks on arbitrary domains.

4.3 Fine-Tuning Evaluation

Training Details. Table 6 shows the number of hypernym pairs used for the fine-tuning experiment. We use instruction tuning specifically tailored towards taxonomy discovery. In the prefix prompt documented in Listing 1.4, given two concepts A and B , the model must return “True” if A is a subclass of B and “False” otherwise. We train the model on 20 epochs, with a batch size of 4 and a learning rate of $3 \cdot 10^{-6}$. For efficient training, we quantize the model on 4 bits and use the LoRA [10] method with parameters $\alpha = 256$ and $r = 1024$.

Metrics. Similarly to the first experiment, we use Precision, Recall and F1-score to evaluate the performance of the model on the testing set.

Results. The results of the fine-tuning experiment are reported in Table 7. Two general observations can be made about the experiment. Firstly, we notice that fine-tuning drastically improves the task-specific performance of the LLM regardless the domain and corpus version. For instance, while Falcon-7B seems to be initially worse-performing than Zephyr-7B- β overall, the F1-score improves up to almost threefold (real WN-music dataset, from 0.327 to 0.874). Although not surprising for real corpora, an improvement on gibberish corpora is non-trivial: the LLMs show signs of adaptation on gibberish corpora with improved

Table 7. Taxonomy discovery results with fine-tuning in the GT(X) vs X setting, where X is either **en** or **gib**. Evaluation is performed before and after fine-tuning.

Model (X)	When	WN-sweets			WN-football			WN-music		
		Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Falcon-7B (en)	Before	.564	.507	.388	.561	.504	.325	.631	.509	.327
	After	.923	.902	.907	.867	.871	.868	.874	.881	.874
Falcon-7B (gib)	Before	.275	.491	.352	.594	.501	.309	.314	.498	.300
	After	.725	.663	.655	.685	.687	.679	.708	.703	.683
Zephyr-7B- β (en)	Before	.898	.845	.853	.772	.679	.618	.783	.722	.674
	After	.905	.906	.897	.940	.939	.939	.941	.939	.940
Zephyr-7B- β (gib)	Before	.746	.572	.500	.740	.621	.535	.723	.589	.479
	After	.840	.816	.796	.859	.810	.817	.839	.846	.839

performance. Secondly, while the performance of a model on a gibberish corpus increases after fine-tuning, it never matches the performance of the same model fine-tuned on the real counterpart of the corpus. It is worth noting that this is a limitation of adaptation solely due to the reliance on prior semantics, since a corpus and its gibberish counterpart only differ in their input-label mapping.

Transfer Learning. In spite of the two previous observations, we can hypothesize that the improvement in performance on the gibberish corpora may indicate signs of reasoning and generalization on unseen domains, given that gibberish words are most likely not contained in the vocabulary of LLMs. To elaborate this claim, we propose another experiment: we use Zephyr-7B- β , trained for taxonomy discovery in a domain, and test it on another domain. By only using gibberish corpora, we ensure that most of the domain-specific terminology is anonymized, preventing the LLM from effectively using prior semantics. Results are reported in Fig. 3. We observe the following. Firstly, the F1-scores generally tend to increase after transfer, with the exception of the WN-music to WN-sweets case. The increase in F1-score is substantial, from 14% (WN-football to WN-sweets), up to 32% (WN-music to WN-football). This result is quite promising, as it points towards the possibility of OL on arbitrary domains with LLMs with effective pre-training. Secondly, the precision tends to drop in favor of the recall, indicating that fine-tuning makes the LLM more sensitive to syntactic clues of hypernymy relations at the cost of making them less precise. Due to the fact that both the training and the testing domains are made of gibberish words, we attribute the performance improvements of the LLM (with respect to its base version) to emerging reasoning capabilities: the fine-tuning LLM becomes more capable of abstraction and of focusing on semantic relationships between concepts, rather than the concepts themselves.

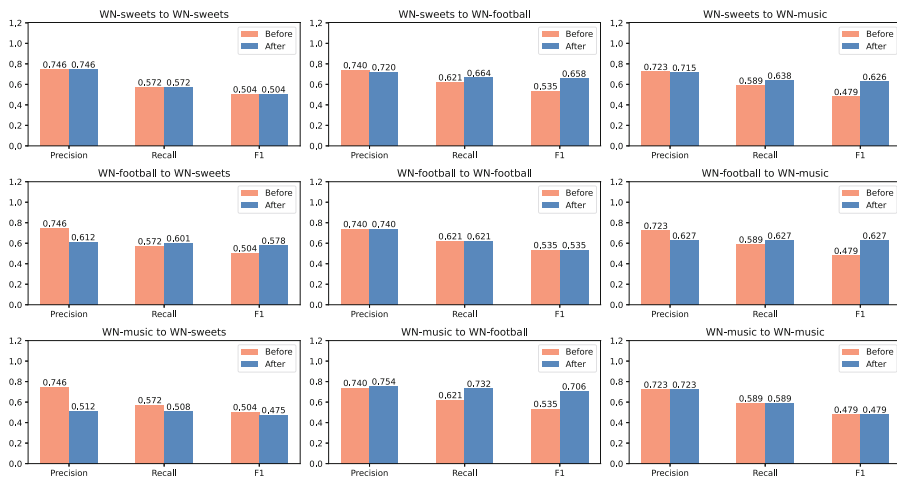


Fig. 3. Domain transfer experiment results with Zephyr-7B- β . Only gibberish corpora are selected for both training and testing.

5 Conclusion

We have explored and tested the limits of adaptability and generalizability of LLMs, and observed that LLMs do not adapt well to arbitrary domains. By creating gibberish datasets based on real data and real domains from WordNet, and using LLMs to perform ontology learning tasks on these data, it is realized that LLMs are unable to consistently retrieve the same taxonomic relationships between analogous concepts, which highlights their clear reliance on priorly learned semantics, lexical senses, and the frame of the tokens. However, we notice that after fine-tuning on gibberish data, LLMs improve at discovering hierarchies, both on the domain they were trained on and other arbitrary domains. We attribute this improvement to the emergence of reasoning with lexical semantics. Our work serves as cautionary advice for the community that LLMs do not adapt to arbitrary domains, and we hope that it can inspire future work to leverage reasoning with LLMs for Ontology Learning.

Supplemental Material Statement. Generated datasets (real and gibberish for all domains), source code for generating the synthetic datasets from the Open English WordNet, and for fine-tuning or evaluating LLMs on OL tasks are available online.³

Acknowledgement. The work was partially supported by EU project: enRichMyData (HORIZON-CL4-2021-DATA-01 - GA 101070284).

³ <https://github.com/boschresearch/llm-vs-gibberish-ontologies>.

References

1. Almazrouei, E., et al.: Falcon-40B: an open large language model with state-of-the-art performance (2023)
2. Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Res.* **32 Database issue**, D267–70 (2004)
3. Brown, T.B., et al.: Language models are few-shot learners (2020)
4. Caufield, J.H., et al.: Structured prompt interrogation and recursive extraction of semantics (spires): a method for populating knowledge bases using zero-shot learning (2023)
5. Cheng, D., Huang, S., Wei, F.: Adapting large language models via reading comprehension (2024)
6. GeoNames: Geonames. <https://www.geonames.org/>
7. Giglou, H.B., D'Souza, J., Auer, S.: Llms4ol: large language models for ontology learning (2023)
8. Gururangan, S., et al.: Don't stop pretraining: Adapt language models to domains and tasks. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.acl-main.740>
9. Hertling, S., Paulheim, H.: Olala: ontology matching with large language models. In: *Proceedings of the 12th Knowledge Capture Conference 2023. K-CAP '23*, New York, NY, USA, pp. 131–139. Association for Computing Machinery (2023). <https://doi.org/10.1145/3587259.3627571>
10. Hu, E.J., et al.: Lora: low-rank adaptation of large language models (2021)
11. Kauf, C., et al.: Event knowledge in large language models: the gap between the impossible and the unlikely (2023)
12. Li, J., Cheng, X., Zhao, W.X., Nie, J.Y., Wen, J.R.: Halueval: a large-scale hallucination evaluation benchmark for large language models (2023)
13. Li, X.L., Kuncoro, A., Hoffmann, J., de Masson d'Autume, C., Blunsom, P., Nematzadeh, A.: A systematic investigation of commonsense knowledge in large language models. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirate, pp. 11838–11855. Association for Computational Linguistics (2022). <https://doi.org/10.18653/v1/2022.emnlp-main.812>
14. Liu, H., Singh, P.: Conceptnet - a practical commonsense reasoning tool-kit. *BT Technol. J.* **22**(4), 211–226 (2004). <https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d>
15. Mateiu, P., Groza, A.: Ontology engineering with large language models (2023)
16. McCrae, J.P., Rademaker, A., Bond, F., Rudnicka, E., Fellbaum, C.: English WordNet 2019 – an open-source WordNet for English. In: Vossen, P., Fellbaum, C. (eds.) *Proceedings of the 10th Global Wordnet Conference*, pp. 245–252. Global Wordnet Association, Wroclaw, Poland (2019)
17. Miller, G.A.: Wordnet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995). <https://doi.org/10.1145/219717.219748>
18. Min, S., et al.: Rethinking the role of demonstrations: what makes in-context learning work? (2022)
19. Moskvoretskii, V., Neminova, E., Lobanova, A., Panchenko, A., Nikishina, I.: Taxollama: wordnet-based model for solving multiple lexical semantic tasks (2024)

20. Norouzi, S.S., Mahdaveinejad, M.S., Hitzler, P.: Conversational ontology alignment with chatgpt (2023)
21. OpenAI: Gpt-4 technical report (2024)
22. Petroni, F., et al.: Language models as knowledge bases? (2019)
23. Seitner, J., et al.: A large DataBase of hypernymy relations extracted from the web. In: Calzolari, N., et al. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, pp. 360–367. European Language Resources Association (ELRA) (2016)
24. Shin, S., et al.: On the effect of pretraining corpora on in-context learning by a large-scale language model. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, United States, pp. 5168–5186. Association for Computational Linguistics (2022). <https://doi.org/10.18653/v1/2022.naacl-main.380>
25. Sun, K., Xu, Y.E., Zha, H., Liu, Y., Dong, X.L.: Head-to-tail: How knowledgeable are large language models (llm)? A.K.A. will llms replace knowledge graphs? (2023)
26. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models (2023)
27. Tunstall, L., et al.: Zephyr: direct distillation of LM alignment (2023)
28. wan, Z., Zhang, Y., Wang, Y., Cheng, F., Kurohashi, S.: Reformulating domain adaptation of large language models as adapt-retrieve-revise (2023)
29. Wei, J., et al.: Emergent abilities of large language models (2022)
30. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models (2023)
31. Wei, J., et al.: Symbol tuning improves in-context learning in language models (2023)
32. Wei, J., et al.: Larger language models do in-context learning differently (2023)
33. Zhao, Z., Lee, W.S., Hsu, D.: Large language models as commonsense knowledge for large-scale task planning (2023)