LLMs4OL: Large Language Models for Ontology Learning

Hamed Babaei Giglou(✉), Jennifer D'Souza, and Sören Auer

TIB Leibniz Information Centre for Science and Technology, Hannover, Germany
{hamed.babaei,jennifer.dsouza,auer}@tib.eu

Abstract. We propose the LLMs4OL approach, which utilizes Large Language Models (LLMs) for Ontology Learning (OL). LLMs have shown significant advancements in natural language processing, demonstrating their ability to capture complex language patterns in different knowledge domains. Our LLMs4OL paradigm investigates the following hypothesis: *Can LLMs effectively apply their language pattern capturing capability to OL, which involves automatically extracting and structuring knowledge from natural language text?* To test this hypothesis, we conduct a comprehensive evaluation using the zero-shot prompting method. We evaluate nine different LLM model families for three main OL tasks: term typing, taxonomy discovery, and extraction of non-taxonomic relations. Additionally, the evaluations encompass diverse genres of ontological knowledge, including lexicosemantic knowledge in WordNet, geographical knowledge in GeoNames, and medical knowledge in UMLS.

The obtained empirical results show that foundational LLMs are not sufficiently suitable for ontology construction that entails a high degree of reasoning skills and domain expertise. Nevertheless, when effectively fine-tuned they just might work as suitable assistants, alleviating the knowledge acquisition bottleneck, for ontology construction.

Keywords: Large Language Models · LLMs · Ontologies · Ontology Learning · Prompting · Prompt-based Learning

1 Introduction

Ontology Learning (OL) is an important field of research in artificial intelligence (AI) and knowledge engineering, as it addresses the challenge of knowledge acquisition and representation in a variety of domains. OL involves automatically identifying terms, types, relations, and potentially axioms from textual information to construct an ontology [30]. Numerous examples of human-expert created ontologies exist, ranging from general-purpose ontologies to domain-specific ones, e.g., Unified Medical Language System (UMLS) [9], WordNet [41], GeoNames [53], Dublin Core Metadata Initiative (DCMI) [66], schema.org [20], etc. Traditional ontology creation relies on manual specification by domain experts, which can be time-consuming, costly, error-prone, and impractical when knowledge constantly evolves or domain experts are unavailable. Consequently, OL

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14265, pp. 408–427, 2023.
https://doi.org/10.1007/978-3-031-47240-4_22

techniques have emerged to automatically acquire knowledge from unstructured or semi-structured sources, such as text documents and the web, and transform it into a structured ontology. A quick review of the field shows that traditional approaches to OL are based on lexico-syntactic pattern mining and clustering [2,4,21,23,24,26,28,37,42,54,61,67]. In contrast, recent advances in natural language processing (NLP) through Large Language Models (LLMs) [46] offer a promising alternative to traditional OL methods. The ultimate goal of OL is to provide a cost-effective and scalable solution for knowledge acquisition and representation, enabling more efficient and effective decision-making in a range of domains. To this end, we introduce the LLMs4OL paradigm and empirically ground it as a foundational first step.

Currently, there is no research explicitly training LLMs for OL. Thus to test LLMs for OL for the first time, we made some experimental considerations. The first being: *Do the characteristics of LLMs justify ontology learning?* First, LLMs are trained on extensive and diverse text, similar to domain-specific knowledge bases [51]. This aligns with the need for ontology developers to have extensive domain knowledge. Second, LLMs are built on the core technology of transformers that have enabled their higher language modeling complexity by facilitating the rapid scaling of their parameters. These parameters represent connections between words, enabling LLMs to comprehend the meaning of unstructured text like sentences or paragraphs. Further, by extrapolating complex linguistic patterns from word connections, LLMs exhibit human-like response capabilities across various tasks, as observed in the field of "emergent" AI. This behavior entails performing tasks beyond their explicit training, such as generating executable code, diverse genre text, and accurate text summaries [59,64]. Such ability of LLMs to extrapolate patterns from simple word connections, encoding language semantics, is crucial for OL. Ontologies often rely on analyzing and extrapolating structured information connections, such as term-type taxonomies and relations, from unstructured text [18]. Thus LLMs4OL hypothesis of LLMs' fruitful application for OL appeared conceptually justified.

LLMs are being developed at a rapid pace. At the time of writing of this work, at least 60 different LLMs are reported [5]. This led to our second main experimental consideration. *Which LLMs to test for the LLMs4OL task hypothesis?* Empirical validation of various LLMs is crucial for NLP advancements and selecting suitable models for research tasks. Despite impressive performances in diverse NLP tasks, LLM effectiveness varies. For the foundational groundwork of LLMs4OL, we comprehensively selected eight diverse model families based on architecture and reported state-of-the-art performances at the time of this writing. The three main LLM architectures are encoder, decoder, and encoder-decoder. The selected LLMs for validation are: BERT [16] (encoder-only); BLOOM [56], MetaAI's LLaMA [60], OpenAI's GPT-3 [10], GPT-3.5 [46], GPT-4 [47] (all decoder-only); and BART [33] and Google's Flan-T5 [11] (encoder-decoder). Recent studies show that BERT excels in text classification and named entity recognition [16], BART is effective in text generation and summarization [33], and LLaMA demonstrates high accuracy in various NLP tasks, including reasoning, question answering, and code generation [60]. Flan-T5 emphasizes instruction tuning and exhibits strong multi-task performance [11]. BLOOM's unique multilingual approach achieves robust performance in tasks like text classification and sequence tagging [56]. Lastly, the GPT series stands out for its human-like text generation abilities [10,46,47]. In this work, we aim to comprehensively unify these LLMs for their effectiveness under the LLMs4OL paradigm for the first time.

With the two experimental considerations in place, we now introduce the LLMs4OL paradigm and highlight our contributions. LLMs4OL is centered around the development of ontologies that comprise the following primitives [39]: **1.** a set of strings that describe terminological lexical entries L for conceptual types; **2.** a set of conceptual types T; **3.** a taxonomy of types in a hierarchy H_T; **4.** a set of non-taxonomic relations R described by their domain and range restrictions arranged in a heterarchy of relations H_R; and **5.** a set of axioms A that describe additional constraints on the ontology and make implicit facts explicit. The LLMs4OL paradigm, introduced in this work, addresses three core aspects of OL as tasks, outlined as the following research questions (RQs).

- **RQ1**: *Term Typing Task* – How effective are LLMs for automated type discovery to construct an ontology?
- **RQ2**: *Type Taxonomy Discovery Task* – How effective are LLMs to recognize a type taxonomy i.e. the "is-a" hierarchy between types?
- **RQ3**: *Type Non-Taxonomic Relation Extraction Task* – How effective are LLMs to discover non-taxonomic relations between types?

The diversity of the empirical tests of this work are not only w.r.t. LLMs considered, but also the ontological knowledge domains tested for. Specifically, we test LLMs for lexico-semantic knowledge in WordNet [41], geographical knowledge in GeoNames [1], biomedical knowledge in UMLS [8], and web content type representations in schema.org [48]. For our empirical validation of LLMs4OL, we seize the opportunity to include PubMedBERT [19], a domain-specific LLM designed solely for the biomedical domain and thus applicable only to UMLS. This addition complements the eight domain-independent model families introduced earlier as a ninth model type. Summarily, our main contributions are:

- The LLMs4OL task paradigm as a conceptual framework for leveraging LLMs for OL.
- An implementation of the LLMs4OL concept leveraging tailored prompt templates for zero-shot OL in the context of three specific tasks, viz. term typing, type taxonomic relation discovery, and type non-taxonomic relation discovery. These tasks are evaluated across unique ontological sources well-known in the community. Our code source with templates and datasets per task are released here https://github.com/HamedBabaei/LLMs4OL.
- A thorough out-of-the-box empirical evaluation of eight state-of-the-art domain-independent LLM types (10 models) and a ninth biomedical domain-specific LLM type (11th model) for their suitability to the various OL tasks

considered in this work. Furthermore, the most effective overall LLM is finetuned and subsequently finetuned LLM results are reported for our three OL tasks.

2 Related Work

There are three avenues of related research: ontology learning from text, prompting LLMs for knowledge, and LLM prompting methods or prompt engineering.

Ontology Learning from Text. One of the earliest approaches [23] used lexicosyntactic patterns to extract new lexicosemantic concepts and relations from large collections of unstructured text, enhancing WordNet [41]. WordNet is a lexical database comprising a lexical ontology of concepts (nouns, verbs, etc.) and lexico-semantic relations (synonymy, hyponymy, etc.). Hwang [24] proposed an alternative approach for constructing a dynamic ontology specific to an application domain. The method involved iteratively discovering types and taxonomy from unstructured text using a seed set of terms representing high-level domain types. In each iteration, newly discovered specialized types were incorporated, and the algorithm detected relations between linguistic features. The approach utilized a simple ontology algebra based on inheritance hierarchy and set operations. Agirre et al. [2] enhanced WordNet by extracting topically related words from web documents. This unique approach added topical signatures to enrich WordNet. Kietz et al. [28] introduced the On-To-Knowledge system, which utilized a generic core ontology like GermaNet [22] or WordNet as the foundational structure. It aimed to discover a domain-specific ontology from corporate intranet text resources. For concept extraction and pruning, it employed statistical term frequency count heuristics, while association rules were applied for relation identification in corporate texts. Roux et al. [54] proposed a method to expand a genetics ontology by reusing existing domain ontologies and enhancing concepts through verb patterns extracted from unstructured text. Their system utilized linguistic tools like part-of-speech taggers and syntactic parsers. Wagner [61] employed statistical analysis of corpora to enrich WordNet in non-English languages by discovering relations, adding new terms to concepts, and acquiring concepts through the automatic acquisition of verb preferences. Moldovan and Girju [43] introduced the Knowledge Acquisition from Text (KAT) system to enrich WordNet's finance domain coverage. Their method involved four stages: (1) discovering new concepts from a seed set of terms, expanding the concept list using dictionaries; (2) identifying lexical patterns from new concepts; (3) discovering relations from lexical patterns; and (4) integrating extracted information into WordNet using a knowledge classification algorithm. In [4], an unsupervised method is presented to enhance ontologies with domain-specific information using NLP techniques such as NER and WSD. The method utilizes a general NER system to uncover a taxonomic hierarchy and employs WSD to enrich existing synsets by querying the internet for new terms and disambiguating them through cooccurrence frequency. Khan and Luo [26] employed clustering techniques to find new terms, utilizing WordNet for typing. They used the self-organizing tree algorithm [17], inspired by molecular evolution, to establish an ontology hierarchy. Additionally, Xu et al. [67] focused on automatically acquiring domain-specific terms and relations through a TFIDF-based single-word term classifier, a lexico-syntactic pattern finder based on known relations and collocations, and a relation extractor utilizing discovered lexico-syntactic patterns.

Predominantly, the approaches for OL [62] that stand out so far are based on lexico-syntactic patterns for term and relation extraction as well as clustering for type discovery. Otherwise, they build on seed-term-based bootstrapping methods. The reader is referred to further detailed reviews [6,38] on this theme for a comprehensive overall methodological picture for OL. Traditional NLP was defined by modular pipelines by which machines were equipped step-wise with annotations at the linguistic, syntactic, and semantic levels to process text. LLMs have ushered in a new era of possibilities for AI systems that obviate the need for modular NLP systems to understand natural language which we tap into for the first time for the OL task in this work.

Prompting LLMs for Knowledge. LLMs can process and retrieve facts based on their knowledge which makes them good zero-shot learners for various NLP tasks. Prompting LLMs means feeding an input x using a *template function* $f_{prompt}(x)$, a textual string prompt input that has some unfilled slots, and then the LLMs are used to probabilistically fill the unfilled information to obtain a final string x', from which the final output y can be derived [35]. The LAMA: LAnguage Model Analysis [52] benchmark has been introduced as a probing technique for analyzing the factual and commonsense knowledge contained in unidirectional LMs (i.e. Transformer-XL [13]) and bidirectional LMs (i.e. BERT and ELMo [49]) with cloze prompt templates from knowledge triples. They demonstrated the potential of pre-trained language models (PLMs) in probing facts – where facts are taken into account as subject-relation-object triples or question-answer pairs – with querying LLMs by converting facts into a cloze template which is used as an input for the LM to fill the missing token. Further studies extended LAMA by the automated discovery of prompts [25], finetuning LLMs for better probing [3,32,68], or a purely unsupervised way of probing knowledge from LMs [50]. These studies analyzed LLMs for their ability to encode various linguistic and non-linguistic facts. This analysis was limited to predefined facts that reinforce the traditional linguistic knowledge of the LLMs, and as a result do not reflect how concepts are learned by the LLMs. In response to this limitation, Dalvi et al. [14] put forward a proposal to explore and examine the latent concepts learned by LLMs, offering a fresh perspective on BERT. They defined concepts as "a group of words that are meaningful," i.e. that can be clustered based on relations such as lexical, morphological, etc. In another study [55], they propose the framework *ConceptX* by extending their studies on seven LLMs in latent space analysis with the alignment of the grouped concepts to human-defined concepts. These works show that using LLMs and accessing the concept's latent spaces, allows us to group concepts and align them to predefined types and type relations discovery.

Prompt Engineering. As a novel discipline, prompt engineering focuses on designing optimal instructions for LLMs to enable successful task performance. Standard prompting [63] represents a fundamental approach for instructing LLMs. It allows users to craft their own customized "self-designed prompts" to effectively interact with LLMs [10] and prompt them to respond to the given prompt instruction straightaway with an answer. Consider the manually crafted FLAN collection [36] addressing diverse NLP tasks other than OL as an exemplar. Notably, the nature of some problems naturally encompass a step-by-step thought process for arriving at the answer. In other words, the problem to be solved can be decomposed as a series of preceding intermediate steps before arriving at the final solution. E.g., arithmetic or reasoning problems. Toward explainability and providing language models in a sense "time to think" helping it respond more accurately, there are advanced prompt engineering methods as well. As a first, as per the Chain-of-Thought (CoT) [65] prompting method, the prompt instruction is so crafted that the LLM is instructed to break down complex tasks as a series of incremental steps leading to the solution. This helps the LLM to reason step-by-step and arrive at a more accurate and logical conclusion. On the other hand Tree-of-Thoughts (ToT) [69] has been introduced for tasks that require exploration or strategic lookahead. ToT generalizes over CoT prompting by exploring thoughts that serve as intermediate steps for general problem-solving with LLMs. Both CoT and ToT unlock complex reasoning capabilities through intermediate reasoning steps in combination with few-shot or zero-shot [29] prompting. Another approach for solving more complex tasks is using decomposed prompting [27], where we can further decompose tasks that are hard for LLMs into simpler solvable sub-tasks and delegate these to sub-task-specific LLMs.

Given the LLMs4OL task paradigm introduced in this work, complex prompting is not a primary concern, as our current focus is on the initial exploration of the task to identify the areas where we need further improvement. We want to understand how much we have accomplished so far before delving into more complex techniques like CoT, ToT, and decomposed prompting. Once we have a clearer picture of the model's capabilities and limitations in a standard prompting setting, we can then consider other than standard prompt engineering approaches by formulating OL as a stepwise reasoning task.

3 The LLMs4OL Task Paradigm

The Large Language Models for Ontology Learning (LLMs4OL) task paradigm offers a conceptual framework to accelerate the time-consuming and expensive construction of ontologies exclusively by domain experts to a level playing field involving powerful AI methods such as LLMs for high-quality OL results; consequently and ideally involving domains experts only in validation cycles. In theory, with the right formulations, all tasks pertinent to OL fit within the LLMs4OL task paradigm. OL tasks are based on ontology primitives [39], including lexical entries L, conceptual types T, a hierarchical taxonomy of types H_T, non-taxonomic relations R in a heterarchy H_R, and a set of axioms A to describe

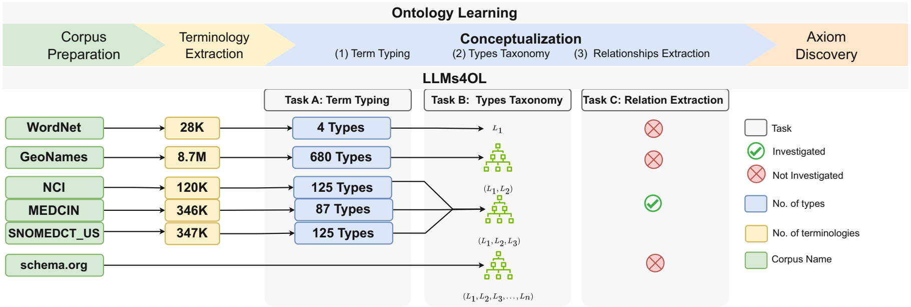

Fig. 1. The LLMs4OL task paradigm is an end-to-end framework for ontology learning in various knowledge domains, i.e. lexicosemantics (WordNet), geography (GeoNames), biomedicine (NCI, MEDICIN, SNOMEDCT), and web content types (schema.org). The three OL tasks empirically validated in this work are depicted within the blue arrow, aligned with the greater LLMs4OL paradigm.(Color figure online)

the ontology's constraints and inference rules. To address these primitives, OL tasks [45] include: 1) Corpus preparation - selecting and collecting source texts for ontology building. 2) Terminology extraction - identifying and extracting relevant terms. 3) Term typing - grouping similar terms into conceptual types. 4) Taxonomy construction - establishing "is-a" hierarchies between types. 5) Relationship extraction - identifying semantic relationships beyond "is-a." 6) Axiom discovery - finding constraints and inference rules for the ontology. This set of six tasks forms the LLMs4OL task paradigm. See Fig. 1 for the proposed LLMs4OL conceptual framework.

In this work, we empirically ground three core OL tasks using LLMs as a foundational basis for future research. However, traditional AI paradigms rely on testing models only on explicitly trained tasks, which is not the case for LLMs. Instead, we test LLMs for OL as an "emergent" behavior [59,64], where they demonstrate the capacity to generate responses on a wide range of tasks despite lacking explicit training. The key to unraveling the emergent abilities of LLMs is to prompt them for their knowledge, as popularized by GPT-3 [10], via carefully designed prompts. As discussed earlier (see Sect. 2), prompt engineering for LLMs is a new AI sub-discipline. In this process, a pre-trained language model receives a prompt, such as a natural language statement, to generate responses without further training or gradient updates to its parameters [35]. Prompts can be designed in two main types based on the underlying LLM pretraining objective: cloze prompts [12,51], which involve filling in blanks in an incomplete sentence or passage per masked language modeling pre-training; and prefix prompts [31,34], which generate text following a given starting phrase and offer more design adaptability to the underlying model. The earlier introduced LLMs4OL paradigm is empirically validated for three select OL tasks using respective prompt functions $f_{prompt}(.)$ suited to each task and model.

Task A - *Term Typing*. A generalized type is discovered for a lexical term.

The generic cloze prompt template is $f^{A}_{c-prompt}(L) := [S?]. [L]\ [P_{domain}]$ *is a* $[MASK]$. where S is an optional context sentence, L is the lexical term prompted for, P_{domain} is a domain specification, and the special $MASK$ token is the type output expected from the model. Since prompt design is an important factor that determines how the LLM responds, eight different prompt template instantiations of the generic template were leveraged with final results reported for the best template. E.g., if WordNet is the base ontology, the part-of-speech type for the lexical term is prompted. In this case, template 1 is "[S]. [L] POS is a [MASK]." Note here "[P_{domain}]" is POS. Template 2 is "[S]. [L] part of speech is a [MASK]." Note here "[P_{domain}]" is "part of speech." In a similar manner, eight different prompt variants from the generic template were created. However, the specification of "[P_{domain}]" depended on the ontology's knowledge domain.

The prefix prompt template reuses the cloze prompt template but appends an additional "instruction" sentence and replaces the special [MASK] token with a blank or "?" symbol. Generically, it is $f^{A}_{p-prompt}(T) = [instruction] + f^{A}_{c-prompt}(T)$, where the instruction is "Perform a sentence completion on the following sentence:" Based on the eight variations created from the generic cloze template prompt, subsequently eight template variations were created for the prefix prompting of the LLMs as well with best template results reported.

Task B - *Taxonomy Discovery*. Here a taxonomic hierarchy between pairs of types is discovered.

The generic cloze prompt template is $f^{B}_{c-prompt}(a, b) := [a|b]$ *is* $[P_{hierarchy}]$ *of* $[b|a]$. *This statement is* $[MASK]$. Where (a, b) or (b, a) are type pairs, $P_{hierarchy}$ indicates superclass relations if the template is initialized for top-down taxonomy discovery, otherwise indicates subclass relations if the template is initialized for bottom-up taxonomy discovery. In Task B, the expected model output for the special [MASK] token for a given type pair was true or false.

Similar to term typing, eight template variations of the generic template were created. Four of which were predicated on the top-down taxonomy discovery. E.g., "[a] is the superclass of [b]. This statement is [MASK]." Note here, $[P_{hierarchy}]$ is "superclass". Other three templates were based on $[P_{hierarchy}] \in$ parent class, supertype, ancestor class. And four more template instantiations predicated on the bottom-up taxonomy discovery were based on $[P_{hierarchy}] \in$ subclass, child class, subtype, descendant class. Thus eight experiments per template instantiation for the applicable LLM were run and the results from the best template were reported.

The prefix prompt template, similarly, reuses the cloze prompt template with the [MASK] token replaced with a blank or "?" symbol. It is $f^{B}_{p-prompt}(a, b) = [instruction] + f^{B}_{c-prompt}(a, b)$, with instruction "Identify whether the following statement is true or false:"

Task C - *Non-Taxonomic Relation Extraction*. This task discovers non-taxonomic semantic heterarchical relations between types.

The cloze prompt template is $f^{C}_{c-prompt}(h, r, t) := [h]$ *is* $[r]\ [t]$. *This statement is* $[MASK]$. Where h is a head type, t is a tail type, and r is a non-taxonomic relationship between h and r. To support the discovery of a heterarchy that can consist of a 1-M relational cardinality, for a given relation, all possible type pairs of the ontology were created. The expected output for the [MASK] token was again true or false. Note, unlike in Task A and B, the given template was used as is and no variations of it were created.

Again, the prefix prompt template reuses the cloze prompt template as the other tasks, with instructions similar to task B. It is $f^{C}_{p-prompt}(h, r, t) = [instruction] + f^{C}_{c-prompt}(h, r, t)$

4 LLMs4OL - Three Ontology Learning Tasks Evaluations

4.1 Evaluation Datasets - Ontological Knowledge Sources

To comprehensively assess LLMs for the three OL tasks presented in the previous section, we cover a variety of ontological knowledge domain sources. Generally, across the tasks, four knowledge domains are represented, i.e. lexicosemantic – WordNet [41], geographical – GeoNames [1], biomedicine – Unified Medical Language System (UMLS) [8] teased out as the National Cancer Institute (NCI) [44], MEDCIN [40], and Systematized Nomenclature of Medicine – Clinical Terms United States (SNOMEDCT_US) [57] subontologies, and content representations in the web – schema.org [48]. Tasks A, B, and C applied only to UMLS. In other words, the ontology has a supporting knowledge base with terms that can be leveraged in the test prompts for term typing as Task A, taxonomic hierarchical relational prompts as Task B, and non-taxonomic heterarchical relational prompts as Task C. The GeoNames source came with a knowledge base of terms instantiated for types and taxonomic relations, therefore, was leveraged in the Task A and B as OL tests with LLMs of this work. The WordNet source could be leveraged only in Task A since it came with an instantiated collection of lexical terms for syntactic types. It was not applicable in the Tasks B and C for OL defined in this work since the semantic relations in WordNet are lexicosemantic, in other words, between terms directly and not their types. Finally, since the schema.org source offered only typed taxonomies as standardized downloads, it was leveraged only in the OL Task B of this work. In this case, we refrained from scraping the web for instantiations of the schema.org taxonomy. For all other ontological knowledge sources considered in this work that were relevant to Task A, the term instantiations were obtained directly from the source. This facilitates replicating our Task A dataset easily. Detailed information on the ontological knowledge sources per task with relevant dataset statistics are presented next.

Task A Datasets. Table 1 shows statistical insights for the Task A dataset where we used terms from WordNet, GeoNames, and UMLS. For WordNet we used the WN18RR data dump [15] that is derived from the original WordNet but released as a benchmark dataset with precreated train and test splits. Overall, it consists of 40,943 terms with 18 different relation types between the terms and four term types (noun, verb, adverb, adjective). We combined the original validation and test sets as a single test dataset. GeoNames comprises 680 categories of geographical locations, which are classified into 9 higher-level categories, e.g. H for stream, lake, and sea, and R for road and railroad. UMLS contains almost three million concepts from various sources which are linked together by semantic relationships. UMLS is unique in that it is a greater semantic ontological network that subsumes other biomedical problem-domain restricted subontologies. We grounded the term typing task to the semantic spaces of three select subontological sources,i.e. NCI, MEDCIN, and SNOMEDCT_US.

Table 1. Task A term typing dataset counts across three core ontological knowledge sources, i.e. WordNet, GeoNames, and UMLS, where for Task A UMLS is represented only by the NCI, MEDCIN, and SNOMEDCT_US subontological sources. The unique term types per source that defined Task A Ontology Learning is also provided.

Parameter	WordNet	GeoNames	NCI	MEDCIN	SNOMEDCT_US
Train Set Size	40,559	8,078,865	96,177	277,028	278,374
Test Set Size	9,470	702,510	24,045	69,258	69,594
Types	4	680	125	87	125

The train datasets were reserved for LLM fine-tuning. Among the 11 models, we selected the most promising one based on its zero-shot performance. The test datasets were used for evaluations in both zero-shot and fine-tuned settings.

Task B Datasets. From GeoNames, UMLS, and schema.org we obtained 689, 127, and 797 term types forming type taxonomies. Our test dataset was constructed as type pairs, where half represented the taxonomic hierarchy while the other half were not in a taxonomy. This is based on the following formulations.

$$\forall(a \in T_n, b \in T_{n+1}) \longmapsto (aRb \wedge b\neg Ra)$$

$$\forall(a \in T_n, b \in T_{n+1}, c \in T_{n+2}); (aRb \wedge bRc) \longmapsto aRc$$
$$\forall(a \in T_n, b \in T_{n+1}, c \in T_{n+2}); (c\neg Rb \wedge b\neg Ra) \longmapsto c\neg Ra$$

Where a, b, and c are types at different levels in the hierarchy. T is a collection of types at a particular level in the taxonomy, where $n+2 > n+1 > n$ and n is the root. The symbol R represents "a is a super class of type b" as a true taxonomic relation. Conversely, the $\neg R$ represents "b is a super class of type a" as a false taxonomic relation. Furthermore, transitive taxonomic relations, $(aRb \wedge bRc) \longmapsto aRc$, were also extracted as true relations, while their converse, i.e. $(c\neg Rb \wedge b\neg Ra) \longmapsto c\neg Ra$ were false relations.

Task C Datasets. As alluded to earlier, Task C evaluations, i.e. non-taxonomic relations discovery, were relegated to the only available ontological knowledge source among those we considered i.e. UMLS. It reports 53 non-taxonomic relations across its 127 term types. The testing dataset comprised all pairs of types for each relation, where for any given relation some pairs are true while the rest are false candidates. Task B and Task C datasets' statistics are in Table 2.

Table 2. Dataset statistics as counts per reported parameter for Task B type taxonomic hierarchy discovery and Task C type non-taxonomic heterarchy discovery across the pertinent ontological knowledge sources respectively per task.

Task	Parameter	GeoNames	UMLS	schema.org
Task B	Types	689	127	797
	Levels	2	3	6
	Positive/Negative Samples	680/680	254/254	2,670/2,670
	Train/Test split	272/1,088	101/407	1,086/4,727
Task C	Non-Taxonomic Relations	–	53	–
	Positive/Negative Samples	–	5,641/1,896	–
	Train/Test Split	–	1,507/6,030	–

4.2 Evaluation Models-Large Language Models (LLMs)

As already introduced earlier, in this work, we comprehensively evaluate eight main types of domain-independent LLMs reported as state-of-the-art for different tasks in the community. They are: BERT [16] as an encoder-only architecture, BLOOM [56], LLaMA [60], GPT-3 [10], GPT-3.5 [46], and GPT-4 [47] as decoder-only models, and finally BART [33] and Flan-T5 [11] as encoder-decoder models. Note these LLMs are released at varying parameter sizes. Thus qualified by the size in terms of parameters written in parenthesis, in all, we evaluate seven LLMs: 1. BERT-Large (340M), 2. BART-Large (400M), 3. Flan-T5-Large (780M), 4. Flan-T5-XL (3B), 5. BLOOM-1b7 (1.7B), 6. BLOOM-3b (3B), 7. GPT-3 (175B), 8. GPT-3.5 (174B), 9. LLaMA (7B), and GPT-4 (>1T). Additionally, we also test an eleventh biomedical domain-specific model PubMedBERT [19].

In this work, since we propose the LLMs4OL paradigm for the first time, in a sense postulating OL as an emergent ability of LLMs, it is important for us to test different LLMs on the new task. Evaluating different LLMs supports: 1) Performance comparison - this allows us to identify which models are effective for OL, 2) Model improvement - toward OL one can identify areas where the models need improvement, and 3) Research advancement - with our results from testing and comparing different models, researchers interested in OL could potentially identify new areas of research and develop new techniques for improving LLMs.

4.3 Evaluations

Metrics. Evaluations for Task A are reported as the mean average precision at k (MAP@K), where k = 1, since this metric was noted as being best suited to the task. Specifically, in our case, for term typing, MAP@1 measures the average precision of the top-1 ranked term types returned by an LLM for prompts initialized with terms from the evaluation set. And evaluations for Tasks B and C are reported in terms of the standard F1-score based on precision and recall.

Results-Three Ontology Learning Tasks Zero-Shot Evaluations. The per task overall evaluations are reported in Table 3. The three main rows of the table marked by alphabets A, B, and C correspond to term typing, type taxonomy discovery, and type non-taxonomic relational hetrarchy discovery results, respectively. The five subrows against Task A shows term typing results for WordNet, GeoNames, and the three UMLS subontologies, viz. NCI, SNOMEDCT_US, and MEDCIN. The three subrows against Task B shows type taxonomy discovery results for GeoNames, UMLS, and schema.org, respectively. Task C evaluation results are provided only for UMLS. We first examine the results in the zero-shot setting, i.e. for LLMs evaluated out-of-the-box, w.r.t. three RQs.

Table 3. Zero-shot results across 11 LLMs and finetuned Flan-T5-Large and Flan-T5-XL LLMs results reported for ontology learning Task A i.e. term typing in MAP@1, and as F1-score for Task B i.e. type taxonomy discovery, and Task C i.e. type non-taxonomic relation extraction. The results are in percentages.

		Zero-Shot Testing											Finetuned	
Task	Dataset	BERT-Large	PubMedBERT	BART-Large	Flan-T5-Large	Flan-T5-XL	BLOOM-1b7	BLOOM-3b	GPT-3	GPT-3.5	LLaMA-7B	GPT-4	Flan-T5-Large*	Flan-T5-XL*
A	*WordNet*	27.9	–	2.2	31.3	52.2	79.2	79.1	37.9	**91.7**	81.4	90.1	76.9	**86.3**
	GeoNames	38.3	-	23.2	13.2	33.8	28.5	28.8	22.4	35.0	29.5	**43.3**	16.9	18.4
	NCI	11.1	5.9	9.9	9.0	9.8	12.4	15.6	12.7	14.7	7.7	**16.1**	31.9	**32.8**
	SNOMEDCT_US	21.1	28.5	19.8	24.3	31.6	37.0	**37.7**	24.4	25.0	13.8	27.8	33.4	**43.4**
	MEDCIN	8.7	15.6	12.7	13.0	18.5	28.8	**29.8**	25.7	23.9	4.9	23.7	38.4	**51.8**
B	*GeoNames*	54.5	–	55.4	59.6	52.4	36.7	48.3	53.2	**67.8**	33.5	55.4	62.5	59.1
	UMLS	48.2	33.7	49.9	55.3	64.3	38.3	37.5	51.6	70.4	32.3	**78.1**	53.4	**79.3**
	schema.org	44.1	–	52.9	54.8	42.7	48.6	51.3	51.0	**74.4**	33.8	74.3	91.7	**91.7**
C	*UMLS*	40.1	42.7	42.4	46.0	**49.5**	43.1	42.7	38.8	37.5	20.3	41.3	49.1	**53.1**

RQ1: How Effective are LLMs for Task A, i.e. Automated Type Discovery?. We examine this question given the results in 5 subrows against the row A, i.e. corresponding to the various ontological datasets evaluated for Task A. Of the five ontological sources, the highest term typing results were achieved on the 4-typed WordNet at 91.7% MAP@1 by GPT-3.5. This high performance can be attributed in part to the simple type space of WordNet with only 4 types.

However, looking across the other LLMs evaluated on WordNet, in particular even GPT-3, scores in the range of 30% MAP@1 seem to be the norm with a low of 2.2% by BART-Large. Thus LLMs that report high scores on WordNet should be seen as more amenable to syntactic typing regardless of the WordNet simple type space. Considering all the ontological sources, Geonames presents the most fine-grained types taxonomy of 680 types. Despite this, the best result obtained on this source is 43.3% from GPT-4 with BERT-Large second at 38.3%. This is better than the typing evaluations on the three biomedical datasets. Even the domain-specific PubMedBERT underperforms. In this regard, domain-independent models with large-scale parameters such a BLOOM (3B) are more amenable to this complex task. Since biomedicine entails deeper domain-specific semantics, we hypothesize better performance not just from domain-specific fine-tuning but also strategically for task-specific reasoning.

The results overview is: 91.7% WordNet by GPT-3.5 > 43.3% GeoNames by GPT-4 > 37.7% SNOMEDCT_US by BLOOM-3b > 29.8% MEDCIN by BLOOM-3b > 16.1% NCI by GPT-4.

Notably this work addresses Task A as a text generation task for the term types. We wish to highlight that Task A can alternatively be tackled as a classification task. For instance, given the set of types for Task A: WordNet - 4, GeoNames - 680, NCI - 125, MEDICIN - 87, and SNOMED_CT - 125, the task can be respectively formulated as a multiclass classification task. We anticipate the classification task complexity to grow with the number of classes. Generally, our only reservation here is that the set of types needs to be known in advance. By following the LLM generation approach instead, we allow the LLM to generate the closest class it thinks applicable and in this work then evaluate how close its generated class is to the one the human-annotated or typed for the term.

RQ2: How Effective are LLMs to Recognize a Type Taxonomy i.e. the "is-a" Hierarchy Between Types?. We examine this question given the results in the 3 subrows against the main row B, i.e. corresponding to the three ontological sources evaluated for Task B. The highest result was achieved for UMLS by GPT-4 at 78.1%. Of the open-source models, Flan-T5-XL achieved the best result at 64.3%. Thus for term taxonomy discovery, LLMs on average have proven most effective in the zero-shot setting on the biomedical domain.

The results overview is: 78.1% UMLS by GPT-4 > 74.4% schema.org by GPT-3.5 > 67.8% GeoNames by GPT-3.5. Note the three GPT models were not open-sourced and thus we tested them with a paid subscription. For the open-source models, the results overview is: 64.3% UMLS by Flan-T5-XL > 59.6% GeoNames by Flan-T5-XL > 54.8% schema.org by Flan-T5-Large.

RQ3: How Effective are LLMs to Discover Non-Taxonomic Relations Between Types?. We examine this question given the results in Table 3 row for Task C, i.e. for UMLS. The best result achieved is 49.5% by Flan-T5-XL. We consider this a fairly good result over a sizeable set of 7,537 type pairs that are in true non-taxonomic relations or are false pairs.

Finally, over all the three tasks considered under the LLMs4OL paradigm, term typing proved the hardest obtaining the lowest overall results for most of its ontological sources tested including the biomedical domain in particular. Additionally in our analysis, GPT, Flan-T5, and BLOOM variants showed improved scores with increase in parameters, respectively. This held true for the closed-sourced GPT models, i.e. GPT-3 (175B) and GPT-3.5 (175B) to GPT-4 (>1T) and the open-sourced models, i.e. Flan-T5-Large (780M) to Flan-T5-XL (3B) and BLOOM from 1.7B to 3B. Thus it seems apparent that with an increased number of LLM parameters, we can expect an improvement in ontology learning.

Note, UMLS offers a robust empirical foundation for Task C. In future work, we propose ConceptNet [58] encompassing commonsense knowledge facts and DBpedia [7] encompassing general knowledge on wide range of topics, including but not limited to geography, history, science, literature, arts, and sports.

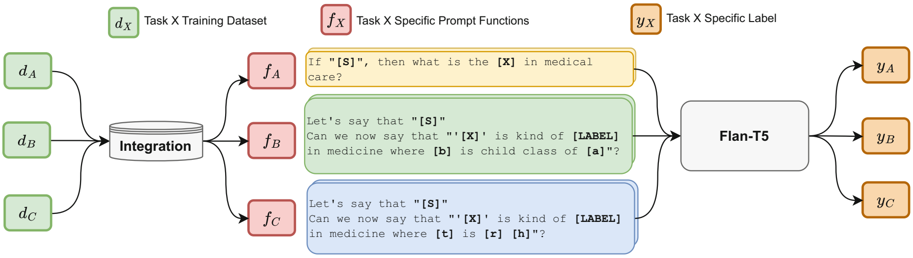
d_X Task X Training Dataset
f_X Task X Specific Prompt Functions
y_X Task X Specific Label
d_A
d_B
d_C
Integration
f_A
f_B
f_C
If "[S]", then what is the [X] in medical care?
Let's say that "[S]" Can we now say that "'[X]' is kind of [LABEL] in medicine where [b] is child class of [a]"?
Let's say that "[S]" Can we now say that "'[X]' is kind of [LABEL] in medicine where [t] is [r] [h]"?
Flan-T5
y_A
y_B
y_C

Fig. 2. An illustration of the LLM finetuning workflow on tasks for ontology learning.

Results-Three Ontology Learning Tasks Finetuned LLM Evaluations. Our zero-shot test results indicate that while LLMs seem promising for OL they would need task-specific finetuning to be a practically viable solution. To this end, we adopt the method of "instruction tuning" proposed as the FLAN collection which is the only known systematically deconstructed, effective way to finetune LLMs [36]. For finetuning, we choose the Flan-T5 LMM for two reasons: 1) it is open-source: we intend to foster future research directions for models unhidden behind paywalls to aid in democratizing LLM research, and 2) it showed consistently good performance across all tasks. The finetuning instructions were instantiated from a small selection of eight samples of each knowledge source' reserved training set and fed in a finetuning workflow shown in Fig. 2. The finetuned Flan models' results (see last two columns in Table 3) are significantly boosted across almost all tasks. For task A, we observed an average improvement of 25% from zero-shot to the finetuned model for both Flan-T5 variants. Notably, SNOMEDCT_US showed least improvement of 9%, while the WordNet showed the most improvement of 45%. For task B we marked an average improvement of 18%, and for task C 3%. Given an illustration of the results in Fig. 3 shows that on average finetuned models, even with fewer parameters outperforms models with 1000x or more parameters across the three OL tasks. These insights appear crucial to expedite developmental research progress for practical tools for OL using LLMs which we plan to leverage in our future work.

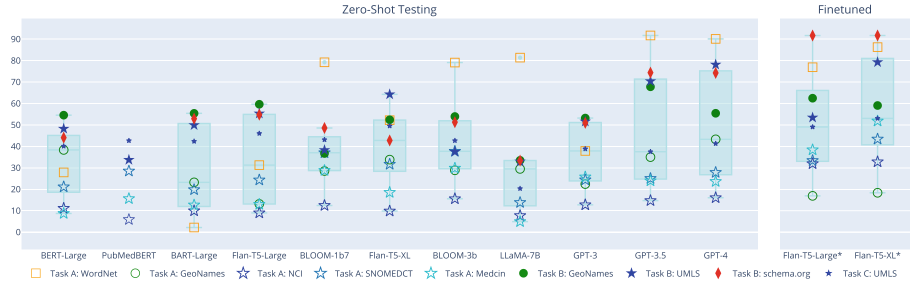

Fig. 3. Comparative visual of the zero-shot and finetuned results. Unfilled shapes, filled shapes, and small filled stars represent performances in tasks A, B, and C, respectively.

5 Error Analysis and Limitations

For error analysis, our evaluation results are summarized in Fig. 3 depicting comparable model ranks.

LLaMA-7B Overall Low Performance. For Task A, the model produced code or prompts instead of correct responses. It performed relatively better on GeoNames due to term types present in location names (e.g.,"Huggins Church" includes the type "church"). For Tasks B and C, it exhibited a bias towards the false category. A limitation in our experiments is the absence of few-shot testing. We hypothesize that models like LLaMA can achieve better performance if shown task examples within the prompt.

Dataset Specific Error Analysis (WN18RR) – Task A. WordNet consists of ≈7k nouns, ≈2K verbs, and ≈0.4K rest of POS tags (adjective and adverbs). Note LLMs tested for Task A are tested for a generation task, which means they can generate text for types that do not map one-to-one to the gold standard. E.g., the best model i.e. GPT-3.5 for 9k test samples generated 43 distinct texts as types with the most frequent being: noun, verb, noun phrase, and adjective. This points out a second limitation of our work, i.e. the possibility for heuristics-based generated answer set mapping to the gold standard.

BERT, showing among the lowest performance on the task (63% lower than the best), generated 177 different answer texts with "verb" being the most frequent (7k times), followed by: noun, joke, and pun. Thus the BERT-based models, including BART, seem to not grasp the syntactic typing task directly from a zero-shot prompt, thus pointing toward the earlier identified limitation of our work for few-shot tests as the alternative method for better results.

Dataset Specific Error Analysis (NCI) – Task A. Overall, the LLMs are least effective on Task A for the NCI biomedical knowledge source. The best-performing open-source BLOOM-3B LLM generated 4k distinct answer texts for a test set of 24k instances, with the most frequently generated texts being: "protein that is involved in," "drug that is used to," "rare disease," and "common problem." On the other hand, the best-performing closed-sourced GPT-4 model generated 17k different answer texts from the identical test set, with the most frequently generated texts being: "term that does not exist," "term that does not exist or is not recognized in," and "term that does not exist or is not commonly used." Both models show varying proficiency and limitations in the NCI biomedical ontology. The NCI Thesaurus covers cancer-related topics, including diseases, agents, and substances. The low LLM performance could be attributed to high domain specialization. Even domain-specific LLMs like PubMedBERT did not yield promising results, suggesting a need for task-specific training or finetuning. While our finetuning experiments obtained boosted scores offering credence to our hypothesis, a limitation is the low number of training samples used which can be addressed by using a large training set.

6 Conclusions and Future Directions

Various initiatives benchmark LLM performance, revealing new task abilities [59,64]. These benchmarks advance computer science's understanding of LLMs. We explore LLMs' potential for Ontology Learning (OL) [18,39] through our introduced conceptual framework, LLMs4OL. Extensive experiments on 11 LLMs across three OL tasks demonstrate the paradigm's proof of concept. Our codebase facilitates replication and extension of methods for testing new LLMs. Our empirical results are promising to pave future work for OL.

Future research directions in the field of OL with LLMs can focus on several key areas. First, there is a need to enhance LLMs specifically for OL tasks, exploring novel architectures and fine-tuning to capture ontological structures better. Second, expanding the evaluation to cover other diverse knowledge domains would provide a broader understanding of LLMs' generalizability. Third, hybrid approaches that combine LLMs with traditional OL techniques, such as lexico-syntactic pattern mining and clustering, could lead to more accurate and comprehensive ontologies. Fourth, further research can delve into the extraction of specific semantic relations, like part-whole relationships or causality, to enhance the expressiveness of learned ontologies. Standardizing evaluation metrics, creating benchmark datasets, exploring dynamic ontology evolution, and domain-specific learning are important directions. Additionally, integrating human-in-the-loop approaches with expert involvement would enhance ontology relevance and accuracy. Exploring these research directions will advance LLM-based OL, enhancing knowledge acquisition and representation across domains.

Supplemental Material Statement: Our LLM templates, detailed results, and codebase are publicly released as supplemental material on Github https://github.com/HamedBabaei/LLMs4OL.

Acknowledgements. We thank the anonymous reviewers for their detailed and insightful comments on an earlier draft of the paper. This work was jointly supported by the German BMBF project SCINEXT (ID 01lS22070), DFG NFDI4DataScience (ID 460234259), and ERC ScienceGraph (ID 819536).

Author contributions. Hamed Babaei Giglou: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft, Visualization. Jennifer D'Souza: Conceptualization, Methodology, Investigation, Resources, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. Sören Auer: Conceptualization, Methodology, Investigation, Resources, Review & Editing, Supervision, Project administration, Funding acquisition.

References

1. Geonames geographical database (2023). http://www.geonames.org/
2. Agirre, E., Ansa, O., Hovy, E., Martínez, D.: Enriching very large ontologies using the www. In: Proceedings of the First International Conference on Ontology Learning, vol. 31. pp. 25–30 (2000)
3. Akkalyoncu Yilmaz, Z., Wang, S., Yang, W., Zhang, H., Lin, J.: Applying BERT to document retrieval with birch. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pp. 19–24. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-3004, https://aclanthology.org/D19-3004
4. Alfonseca, E., Manandhar, S.: An unsupervised method for general named entity recognition and automated concept discovery. In: Proceedings of the 1st International Conference on General WordNet, Mysore, India, pp. 34–43 (2002)
5. Amatriain, X.: Transformer models: an introduction and catalog. arXiv preprint arXiv:2302.07730 (2023)
6. Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M.: A survey of ontology learning techniques and applications. Database 2018, bay101 (2018)
7. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
8. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32**(suppl_1), D267–D270 2004). https://doi.org/10.1093/nar/gkh061
9. Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. Nucleic Acids Res. **32**(suppl_1), D267–D270 (2004)
10. Brown, T.B., et al.: Language models are few-shot learners (2020)
11. Chung, H.W., et al.: Scaling instruction-finetuned language models (2022)
12. Cui, L., Wu, Y., Liu, J., Yang, S., Zhang, Y.: Template-based named entity recognition using bart. arXiv preprint arXiv:2106.01760 (2021)
13. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: attentive language models beyond a fixed-length context (2019)
14. Dalvi, F., Khan, A.R., Alam, F., Durrani, N., Xu, J., Sajjad, H.: Discovering latent concepts learned in BERT. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=POTMtpYI1xH
15. Dettmers, T., Pasquale, M., Pontus, S., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the 32th AAAI Conference on Artificial Intelligence, pp. 1811–1818 (February 2018), https://arxiv.org/abs/1707.01476
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)

17. Dopazo, J., Carazo, J.M.: Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. J. Mol. Evol. **44**(2), 226–233 (1997)
18. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum Comput Stud. **43**(5–6), 907–928 (1995)
19. Gu, Y., et al.: Domain-specific language model pretraining for biomedical natural language processing. ACM Trans. Comput. Healthcare (Health) **3**(1), 1–23 (2021)
20. Guha, R.V., Brickley, D., Macbeth, S.: Schema. org: evolution of structured data on the web. Commun. ACM **59**(2), 44–51 (2016)
21. Hahn, U., Markó, K.G.: Joint knowledge capture for grammars and ontologies. In: Proceedings of the 1st International Conference on Knowledge Capture, pp. 68–75 (2001)
22. Hamp, B., Feldweg, H.: Germanet-a lexical-semantic net for german. In: Automatic Information Extraction and Building Of Lexical Semantic Resources for NLP Applications (1997)
23. Hearst, M.A.: Automated discovery of wordnet relations. WordNet: an electronic lexical database, vol. 2 (1998)
24. Hwang, C.H.: Incompletely and imprecisely speaking: using dynamic ontologies for representing and retrieving information. In: KRDB, vol. 21, pp. 14–20. Citeseer (1999)
25. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? Trans. Asso. Comput. Ling. **8**, 423–438 (2020). https://doi.org/10.1162/tacl_a_00324, https://aclanthology.org/2020.tacl-1.28
26. Khan, L., Luo, F.: Ontology construction for information selection. In: 14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings, pp. 122–127. IEEE (2002)
27. Khot, T., et al.: Decomposed prompting: A modular approach for solving complex tasks (2023)
28. Kietz, J.U., Maedche, A., Volz, R.: A method for semi-automatic ontology acquisition from a corporate intranet. In: EKAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France (October 2000)
29. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners (2023)
30. Konys, A.: Knowledge repository of ontology learning tools from text. Proc. Comput. Sci. **159**, 1614–1628 (2019)
31. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
32. Levy, O., Seo, M., Choi, E., Zettlemoyer, L.: Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 333–342. Association for Computational Linguistics, Vancouver, Canada (Aug 2017). https://doi.org/10.18653/v1/K17-1034, https://aclanthology.org/K17-1034
33. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019)
34. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
35. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comput. Surv. **55**(9) (2023). https://doi.org/10.1145/3560815
36. Longpre, S., et al.: The flan collection: designing data and methods for effective instruction tuning. arXiv preprint arXiv:2301.13688 (2023)

37. Lonsdale, D., Ding, Y., Embley, D.W., Melby, A.: Peppering knowledge sources with salt: Boosting conceptual content for ontology generation. In: Proceedings of the AAAI Workshop on Semantic Web Meets Language Resources, Edmonton, Alberta, Canada (2002)
38. Lourdusamy, R., Abraham, S.: A survey on methods of ontology learning from text. In: Jain, L.C., Peng, S.-L., Alhadidi, B., Pal, S. (eds.) ICICCT 2019. LAIS, vol. 9, pp. 113–123. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38501-9_11
39. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intell. Syst. **16**(2), 72–79 (2001)
40. Medicomp Systems: MEDCIN (January 2023). https://medicomp.com
41. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
42. Missikoff, M., Navigli, R., Velardi, P.: The usable ontology: an environment for building and assessing a domain ontology. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 39–53. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-48005-6_6
43. Moldovan, D.I., GiRJU, R.C.: An interactive tool for the rapid development of knowledge bases. Inter. J. Artifi. Intell. Tools **10**(01n02), 65–86 (2001)
44. National Cancer Institute, National Institutes of Health: NCI Thesaurus (September 2022). http://ncit.nci.nih.gov
45. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: A guide to creating your first ontology (2001)
46. OpenAI: Chatgpt (2023). https://openai.com/chat-gpt/ (Accessed 5 May 2023)
47. OpenAI: Gpt-4 technical report (2023)
48. Patel-Schneider, P.F.: Analyzing Schema.org. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 261–276. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_17
49. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1202, https://aclanthology.org/N18-1202
50. Petroni, F., et al.: How context affects language models' factual predictions. In: Automated Knowledge Base Construction (2020). https://openreview.net/forum?id=025X0zPfn
51. Petroni, F., et al.: Language models as knowledge bases? arXiv preprint arXiv:1909.01066 (2019)
52. Petroni, F., et al.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics (2019)
53. Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., Weikum, G.: YAGO: a multilingual knowledge base from wikipedia, wordnet, and geonames. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 177–185. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_19
54. Roux, C., Proux, D., Rechenmann, F., Julliard, L.: An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions. In: ECAI Workshop on Ontology Learning (2000)

55. Sajjad, H., Durrani, N., Dalvi, F., Alam, F., Khan, A.R., Xu, J.: Analyzing encoded concepts in transformer language models (2022)
56. Scao, T.L., et al.: Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022)
57. SNOMED International: US Edition of SNOMED CT (March 2023). https://www.nlm.nih.gov/healthit/snomedct/us_edition.html
58. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the AAAI conference on Artificial Intelligence. vol. 31 (2017)
59. Srivastava, A., et al.: Beyond the imitation game: quantifying and extrapolating the capabilities of language models. arXiv preprint arXiv:2206.04615 (2022)
60. Touvron, H., et al.: Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
61. Wagner, A.: Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In: ECAI Workshop on Ontology Learning, vol. 61. Citeseer (2000)
62. Wątróbski, J.: Ontology learning methods from text-an extensive knowledge-based approach. Proc. Comput. Sci. **176**, 3356–3368 (2020)
63. Wei, J., et al.: Finetuned language models are zero-shot learners. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=gEZrGCozdqR
64. Wei, J., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022)
65. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems, vol. 35, pp. 24824–24837. Curran Associates, Inc. (2022). https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf
66. Weibel, S.L., Koch, T.: The dublin core metadata initiative. D-lib magazine **6**(12), 1082–9873 (2000)
67. Xu, F., Kurz, D., Piskorski, J., Schmeier, S.: A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In: LREC (2002)
68. Yang, W., Zhang, H., Lin, J.: Simple applications of bert for ad hoc document retrieval (2019)
69. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: deliberate problem solving with large language models (2023)