LLMs4OL 2024: The 1st Large Language Models for Ontology Learning Challenge at the 23rd ISWC
LLMs4OL 2024 Task Participant Papers
https://doi.org/10.52825/ocp.v4i.2482

Published: 02 Oct. 2024

Phoenixes at LLMs4OL 2024 Tasks A, B, and C: Retrieval Augmented Generation for Ontology Learning

Mahsa Sanaei[1], Fatemeh Azizi[1], and Hamed Babaei Giglou[2]

[1]University of Tabriz, Tabriz, Iran
mahsa.san75@gmail.com, fatemeazizii896@gmail.com
[2]TIB Leibniz Information Centre for Science and Technology, Hannover, Germany
hamed.babaei@tib.eu

*Correspondence: Mahsa Sanaei, mahsa.san75@gmail.com

Abstract: Large language models (LLMs) showed great capabilities in ontology learning (OL) where they automatically extract knowledge from text. In this paper, we proposed a Retrieval Augmented Generation (RAG) formulation for three different tasks of ontology learning defined in the LLMs4OL Challenge at ISWC 2024. For task A - term typing - we considered terms as a query and encoded the query through the Query Encoder model for searching through knowledge base embedding of types embeddings obtained through Context Encoder. Next, using Zero-Shot Prompt template we asked LLM to determine what types are appropriate for a given term within the term typing task. Similarly, for Task B, we calculated the similarity matrix using an encoder-based transformer model, and by applying the similarity threshold we considered only similar pairs to query LLM to identify whatever pairs have the ”is-a” relation between a given type and in a case of having the relationships which one is ”parent” and which one is ”child”. In final, for Task C – non-taxonomic relationship extraction – we combined both approaches for Task A and B, where first using Task B formulation, child-parents are identified then using Task A, we assigned them an appropriate relationship. For the LLMs4OL challenge, we experimented with the proposed framework over 5 subtasks of Task A, all subtasks of Task B, and one subtask of Task C using Mistral-7B LLM.

Keywords: Large Language Models, Ontology Learning, Retrieval Augmented Generation, Term Typing, Taxonomy Discovery, Non-Taxonomic Relationship Extraction

1 Introduction

Ontology Learning (OL) is a critical area in knowledge representation and management, addressing the challenges of acquiring and structuring knowledge from diverse textual sources. With the rapid advancements in Natural Language Processing (NLP), particularly through the emergence of Large Language Models (LLMs), there is a compelling opportunity to enhance OL processes. LLMs have demonstrated remarkable capabilities in understanding and generating human language, making them potential candidates for automating the extraction and organization of knowledge from natural language texts. In the work of Babaei Giglou et al. [1] LLMs4OL paradigm was introduced which investigates the hypothesis: *Can LLMs effectively leverage their language pattern recognition abilities to facilitate ontology learning?* Our approach encompasses a comprehensive evaluation of different LLM families across three primary tasks: term typing, taxonomy discovery, and extraction of non-taxonomic relationships. These tasks are evaluated using diverse ontological knowledge sources, including lexicosemantic knowledge from WordNet, geographical knowledge from GeoNames, and medical knowledge from UMLS. The empirical results from our study reveal that while foundational LLMs may struggle with the reasoning and domain expertise required for effective ontology construction, they can serve as valuable assistants when fine-tuned appropriately. This fine-tuning can alleviate the knowledge acquisition bottleneck often encountered in ontology development.

To systematically explore the capabilities of LLMs in OL, we have structured our research into three distinct tasks as described in LLMs4OL 2024 Challenge [2]:

1. **Task A – Term Typing**: This task involves classifying terms into predefined categories across various domains, such as geographical locations in GeoNames and medical terminologies in UMLS.
2. **Task B – Taxonomy Discovery**: Here, we aim to identify hierarchical relationships between term types, utilizing datasets from GeoNames and Schema.org to establish taxonomic structures.
3. **Task C – Non-Taxonomic Relationship Extraction**: This task focuses on identifying semantic relationships between terms that do not conform to hierarchical structures, with a particular emphasis on medical concepts in UMLS.

The rest of the paper is constructed as follows: In section 2 we refer to some previously conducted works. Then in section 3, we describe our methodology and after reporting the results of the study in section 4, we provide information about datasets we used in our implementations.

2 Related works

The construction of ontologies and knowledge graphs (KGs) has traditionally relied on human domain experts to define entities, establish relationships, and ensure data quality. However, the advent of Large Language Models (LLMs) has introduced promising avenues for automating aspects of this labor-intensive process. In the work of Kommineni et al. [3] proposed a semi-automated pipeline for constructing KGs using open-source LLMs. Their approach involves formulating competency questions (CQs), developing an ontology based on these CQs, and constructing KGs with minimal human involvement. The authors demonstrate the feasibility of their pipeline by creating a KG focused on deep learning methodologies, utilizing scholarly publications. Their findings suggest that while LLMs can significantly reduce the human effort required for KG construction, a human-in-the-loop approach remains essential for evaluating the quality of automatically generated content.

Another study [4] introduces ANGEL, a framework that integrates ontology structures and instructive prompting within LLMs for Named Entity Recognition (NER) data augmentation. This framework addresses the challenge of generating scalable training data while maintaining contextual diversity and label consistency. The experimental results indicate that ANGEL outperforms state-of-the-art methods, showcasing the potential of LLMs to enhance NER model performance, especially in low-resource scenarios. OntoChat is presented as a framework designed to facilitate conversational ontology engineering [5]. By leveraging LLMs, OntoChat supports requirement elicitation, analysis, and testing in large collaborative projects. The framework allows users to interact with a conversational agent to create user stories and extract competency questions, thus streamlining the ontology engineering process. Preliminary evaluations indicate positive feedback from domain experts, although challenges such as biases and the need for enhanced insights into implementation costs remain.

One other work presented SPIRES [6], a knowledge extraction approach that utilizes LLMs for zero-shot learning and schema-conforming query answering. SPIRES recursively interrogates prompts to extract information from input text while adhering to a user-defined knowledge schema. The method demonstrates flexibility and customization, enabling it to perform various tasks without requiring new training data. The results indicate that SPIRES can assist in knowledge curation and validation, significantly improving the efficiency of knowledge base creation. Furthermore, researchers investigate the use of LLMs to generate technical content relevant to the SAPPhIRE model of causality. They present a method for hallucination suppression using Retrieval-Augmented Generation (RAG) to ensure the generated content is accurate and scientifically grounded. The study emphasizes the importance of the context provided to the LLM, demonstrating that different contexts can lead to varying quality in the generated responses. This research aims to build a software tool for generating SAPPhIRE models, highlighting the potential of LLMs in technical knowledge generation [7].

In a study, L.Silva et al. [8] explore the creation of capability ontologies using LLMs. The authors conduct experiments with different prompting techniques and LLMs to generate machine-interpretable models from natural language descriptions. Their findings indicate that even complex capabilities can be accurately modeled, significantly reducing the effort and expertise required for ontology creation. The study also emphasizes the need for semi-automated quality checks to ensure the reliability of the generated ontologies. Yushi Sun and his team also investigated whether traditional knowledge graphs should be replaced by LLMs, particularly regarding their ability to capture specialized taxonomies. The authors introduce TaxoGlimpse, a benchmark for evaluating the performance of LLMs across various taxonomies. Their comprehensive experiments reveal that while LLMs perform well on common taxonomies, they struggle with specialized domains and leaf-level entities. The study suggests future research directions that combine LLMs with traditional taxonomies to create novel neural-symbolic taxonomies [9]. Recent research has started to explore the potential of LLMs in ontology matching (OM) using retrieval augmented generation (RAG), leveraging the vast amount of knowledge encoded in these models to perform more sophisticated and context-aware matching. The LLMs4OM [10] framework represents a significant advancement in this direction. It introduces an approach that employs LLMs for OM tasks through two modules dedicated to retrieval and matching, enhanced by zero-shot prompting across three ontology representations: concept, concept-parent, and concept-children. Comprehensive evaluations using 20 OM datasets from various domains demonstrate that LLMs4OM can match and even surpass the performance of traditional OM systems, particularly in complex matching scenarios using RAG.

The mentioned research collectively highlights the transformative potential of LLMs in ontology and KG construction, offering various methodologies to enhance automation and reduce the reliance on human expertise. However, they also underscore the importance of maintaining human oversight to ensure the accuracy and relevance of the generated content. As the field evolves, future research will likely continue to explore the integration of LLMs in knowledge engineering, addressing existing limitations and enhancing the effectiveness of these technologies.

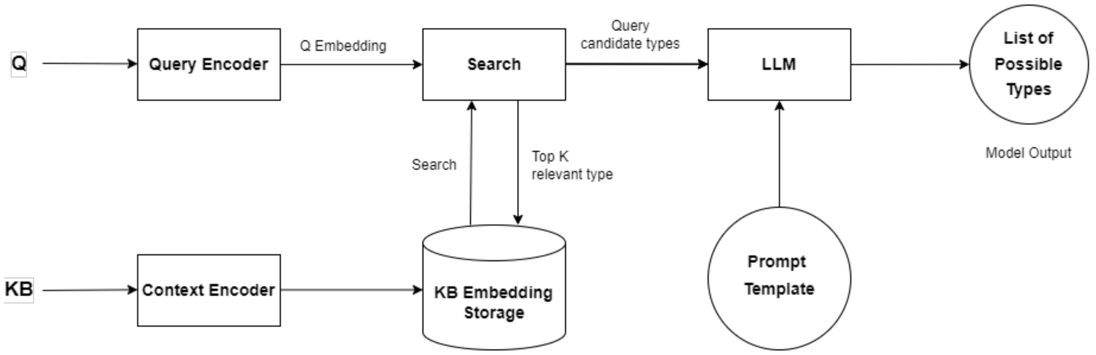

Figure 1. *RAG for Term Typing Task of LLMs4OL*

3 Methodology

3.1 Task A – term typing

In Task A, the goal is to classify terms into predefined categories across various domains. We implemented a Retrieval-Augmented Generation (RAG) approach, leveraging LLMs using QLoRA approach. This setup allowed us to efficiently handle the term classification task without the need for additional fine-tuning. By integrating RAG, we aimed to enhance the accuracy and relevance of the classifications, making it suitable for a wide range of domains where terms might have clear meanings and require exact categorization.

To accomplish the task, the Figure 1 implemented to treat the types as a knowledge base (KB) and employed a Context Encoder model to generate embeddings for these types, which were then stored in the KB Embedding Storage. Specifically, we used the `dpr-ctx_encoder-single-nq-base` model [11], which is a sentence-BERT variant, to create context-aware embeddings. For any given query, we generated the corresponding embedding using a Query Encoder with `dpr-question_encoder-single-nq-base` model [11]. This dual-encoder approach facilitated a robust representation of both terms and types, ensuring that the system could effectively match terms with the most relevant types. Once the embeddings were in place, a Retrieval model searched the KB Embedding Storage to retrieve the top-k candidate types using the cosine similarity metric (we set top-k as 20). These candidate types were then passed to the LLM, specifically the `Mistral-7B-Instruct-v0.3` [12] model, which processed the candidates through a specialized prompt template (as described in Figure 2). The prompt was designed to instruct the LLM to identify the most probable types for the given term and return them in a simple Python list format, without any additional explanation. This process allowed for efficient and accurate term typing, ensuring that the most suitable types were consistently identified for each term.

3.2 Task B - taxonomy discovery

In Task B: Taxonomy Discovery, the focus is on identifying ”is-a” relationships between predefined types, where the goal is to determine the hierarchical child-parent relationships among these types. This process involves analyzing provided types to establish which ones serve as more general categories (parents) and which are more specific instances (children). By uncovering these relationships, we can construct or expand a taxonomy that organizes types in a structured manner, reflecting their inherent hierarchies. The overall workflow for this task is visually summarized in Figure 3.

Given a list of types as a candidate to be assigned to the term, identify the most probable types.

Return types only in the form of a Python list.
Do not provide any explanation.

Term: `<term>`
Candidates: `<candidates-list>`
Suitable types:

Figure 2. *Prompt Template for Task A - Term Typing*

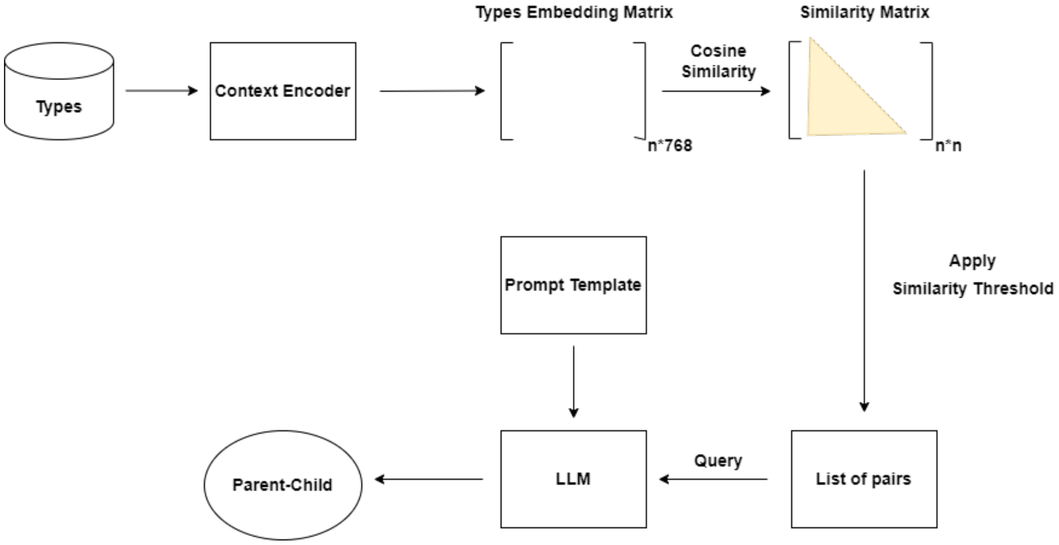

Figure 3. *RAG for Taxonomy Discovery Task of LLMs4OLB*

The first step in this task was to generate a types embedding matrix using a context embedding model. This matrix represents the types in a high-dimensional space, capturing their semantic similarities. To identify potential ”is-a” relationships, we calculated pairwise cosine similarities between all possible pairs of types, producing a cosine-similarity matrix. This matrix serves as the foundation for detecting relationships, with each value representing how closely related two types are in terms of their embeddings. We then applied a threshold-based filter to the lower triangular part of this matrix, effectively narrowing down the list of possible type pairs that might exhibit a child-parent relationship. The filtered pairs were then passed to a Large Language Model (LLM) to assess whether a child-parent relationship exists between each pair. We designed a specific prompt template to guide the LLM in this evaluation. For each pair, the LLM was asked to determine if a hierarchical relationship was present, and if so, to identify which type is the child and which is the parent. The model was instructed to output the results in a JSON format, strictly indicating the child-parent pairs or returning an empty JSON object if no relationship was found. This structured approach ensured that the LLM’s output clear and accurate taxonomy. The prompt template in Figure 4 is used in the taxonomy discover framework.

```
Given two types, determine whether they can have the children-parent relations or not.
Then which one would be a parent and which one would be a child?

Use the following output format:
“‘
{
”child”: ”type”,
”parent”: ”type”
}
“‘
Notes:
- If it is not possible to establish a parent-child relationship.
Just return empty ‘{}‘.
- Do not return anything other than JSON.
- Do not provide any explanation

Term-1: <first-types>
Term-2: <second-type>

###
```

Figure 4. *Prompt Template for Task B - Taxonomy Discovery*

3.3 Task C – Non-Taxonomic Relation Extraction

In Task C: Non-Taxonomic Relation Extraction, the objective is to identify and extract triplets in the form of (head, relation, tail) from a set of given types. These triplets represent non-taxonomic relationships between types, where ”head” and ”tail” are types, and ”relation” defines the nature of their connection. This task leverages the methodologies developed for both Task A (Term Typing) and Task B (Taxonomy Discovery), integrating them to uncover and label relationships beyond simple hierarchical structures.

The first phase of this task mirrors the approach used in Task B: we begin by identifying potential pairs of types that may have a significant relationship, treating the problem similarly to how we discovered ”child-parent” relationships in Task B. We use a context embedding model to create embeddings for the types and then calculate pairwise cosine similarities to determine which pairs are closely related. By applying a threshold to the cosine similarity matrix, we filter out the most promising type pairs, which could potentially form the basis of non-taxonomic triplets.

Once the type pairs are identified, we employ an approach similar to Task A to assign the appropriate relationship (or ”relation”) to each pair, transforming the ”child-parent” identification into a broader relation extraction. The filtered pairs are fed into an LLM using a prompt depicted in Figure 5 to determine the exact nature of the relationship between each pair. The LLM, informed by its understanding of the types, assigns a specific relation to each pair, effectively completing the triplet. This combined approach ensures we can extract meaningful and accurate (head, relation, tail) triplets, providing a comprehensive understanding of the relationships within the given set of types.

```
Given a head and tail type with candidate relations between them, identify the most
probable relation between head and tail.

Notes:
- Return a single relation in the following format:
{'relation':'relation-name'}
- not provide any explanation.

Head-Type: <head-type>
Tail-Type: <tail-type>

Candidate relation between head and tail types: <candid-list>
Suitable relations:
```

Figure 5. *Prompt Template for Task C - Non-Taxonomic Relation Extraction*

4 Results

In the LLMs4OL Challenge, we participated in multiple subtasks across three major tasks: Task A (Term Typing), Task B (Taxonomy Discovery), and Task C (Non-Taxonomic Relation Extraction). Our performance was evaluated based on F1 scores, precision, and recall under both Few-Shot (FS) and Zero-Shot (ZS) testing scenario datasets of the challenge [13]. The results are presented in Table 1.

Table 1. *Phoenixes at LLMs4OL Challenge Results Across LLMs4OL SubTasks.*

SubTasks	**Rank**	**F1**	**Precision**	**Recall**
Task A - Term Typing				
SubTask A.1 (FS) - WordNet	7	0.8158	0.7689	0.8687
SubTask A.3 (FS) - NCI	5	0.0737	0.0562	0.1070
SubTask A.4 (FS) - Cellular Component	5	0.0158	0.0124	0.0217
SubTask A.4 (FS) - Biological Process	5	0.0319	0.0214	0.0622
SubTask A.4 (FS) - Molecular Function	5	0.0700	0.0485	0.1256
Task B - Taxonomy Discovery				
SubTask B.1 (FS) - GeoNames	5	0.0036	0.0019	0.0294
SubTask B.2 (FS) - Schema.org	3	0.0155	0.0079	0.3901
SubTask B.3 (FS) - UMLS	2	0.0960	0.0550	0.3778
SubTask B.4 (FS) - Gene Ontology (GO)	1	0.0164	0.0180	0.0149
SubTask B.5 (FS) - DBpedia Ontology (DPO)	2	0.0164	0.0180	0.0149
SubTask B.6 (ZS) - Food Ontology (FoodOn)	1	0.0308	0.0243	0.0420
Task C - Non-Taxonomic Relationship Extraction				
SubTask C.1 (FS) - UMLS	2	0.0273	0.0433	0.0199

Below, we provide an overview of our results and their insights.

4.1 Task A - Term Typing

In Task A, we participated in five subtasks focused on different ontologies and domains. Our best performance was in *SubTask A.1 (FS) - WordNet*, where we achieved an F1 score of 0.8158. This result indicates a relatively strong ability to classify terms within the WordNet domain, with a precision of 0.7689 and a recall of 0.8687. However, our performance in the other subtasks fell short, particularly in *SubTask A.4 (FS) - Cellular Component*, where we only achieved an F1 score of 0.0158. Similar low scores were observed in *SubTask A.4 (FS) - Biological Process* (F1 = 0.0319) and *SubTask A.4 (FS) - Molecular Function* (F1 = 0.0700). These results suggest that our model struggled with more specialized biological domains, likely due to the complexity and specificity of the terms involved. Overall, the presented results show the formulation of the task with RAG is beneficial, however, fine-tuning is one of the requirements to obtain a better performance as observed in [1].

4.2 Task B - Taxonomy Discovery

In Task B, we explored the discovery of ”is-a” relationships across various ontologies. Our best result was in *SubTask B.3 (FS) - UMLS*, where we ranked 2nd with an F1 score of 0.0960. However, the F1 scores across other subtasks, such as *SubTask B.1 (FS) - GeoNames* (F1 = 0.0036) and *SubTask B.2 (FS) - Schema.org* (F1 = 0.0155), indicate difficulties in accurately identifying taxonomic relationships in these domains. For *SubTask B.3 (FS) - UMLS* the recall score of 0.3778 shows that our approach was competitive in identifying complex relationships within the UMLS domain, however, LLM failed to find appropriate relations.

4.3 Task C - Non-Taxonomic Relationship Extraction

For Task C, we participated in *SubTask C.1 (FS) - UMLS*, which focused on extracting non-taxonomic relationships. Our model achieved an F1 score of 0.0273, ranking 2nd in this subtask. Despite the relatively low F1 score, this result shows that our approach was competitive in identifying complex relationships within the UMLS domain. The precision of 0.0433 and recall of 0.0199 indicate that while our model was able to correctly identify some relationships, there were challenges in capturing the full range of relevant relations, suggesting areas for further improvement.

5 Conclusion

In conclusion, our participation in the LLMs4OL Challenge revealed strengths in certain domains, particularly in Task A for WordNet and in Task B for Food Ontology. However, the generally low F1 scores across many subtasks highlight the challenges of term typing, taxonomy discovery, and relation extraction in highly specialized domains. These results suggest that while our approach has potential, there is significant room for improvement, particularly in enhancing the model’s adaptability to diverse and complex ontologies. The implementation of this work is published in the GitHub repository for the research community at https://github.com/MahsaSanaei/Phoenixes-LLMs4OL-ISWC.

Authors Contributions

Mahsa Sanaei: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft.
Fatemeh Azizi: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft.
Hamed Babaei Giglou: Conceptualization, Investigation, Review & Editing, Supervision.

Competing interests

The authors declare that they have no competing interests.

References

[1] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, *et al.*, Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.

[2] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol 2024 overview: The 1st large language models for ontology learning challenge," *Open Conference Proceedings*, vol. 4, Oct. 2024.

[3] V. K. Kommineni, B. König-Ries, and S. Samuel, *From human experts to machines: An llm supported approach to ontology and knowledge graph construction*, 2024. arXiv: 2403.08345 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2403.08345.

[4] Z. Luo, Y. Wang, W. Ke, R. Qi, Y. Guo, and P. Wang, "Boosting llms with ontology-aware prompt for ner data augmentation," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12 361–12 365. DOI: 10.1109/ICASSP48485.2024.10446860.

[5] B. Zhang, V. A. Carriero, K. Schreiberhuber, *et al.*, *Ontochat: A framework for conversational ontology engineering using language models*, 2024. arXiv: 2403.05921 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2403.05921.

[6] J. H. Caufield, H. Hegde, V. Emonet, *et al.*, *Structured prompt interrogation and recursive extraction of semantics (spires): A method for populating knowledge bases using zero-shot learning*, 2023. arXiv: 2304.02711 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2304.02711.

[7] K. Bhattacharya, A. Majumder, and A. Chakrabarti, *A study on effect of reference knowledge choice in generating technical content relevant to sapphire model using large language model*, 2024. arXiv: 2407.00396 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2407.00396.

[8] L. M. V. da Silva, A. Köcher, F. Gehlhoff, and A. Fay, *On the use of large language models to generate capability ontologies*, 2024. arXiv: 2404.17524 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2404.17524.

[9] Y. Sun, H. Xin, K. Sun, *et al.*, *Are large language models a good replacement of taxonomies?* 2024. arXiv: 2406.11131 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.11131.

[10] H. B. Giglou, J. D'Souza, F. Engel, and S. Auer, *Llms4om: Matching ontologies with large language models*, 2024. arXiv: 2404.10317 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2404.10317.

[11] V. Karpukhin, B. Oguz, S. Min, *et al.*, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.550.

[12] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, *Mistral 7b*, 2023. arXiv: 2310.06825 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.06825.

[13] H. Babaei Giglou, J. D'Souza, S. Sadruddin, and S. Auer, "Llms4ol 2024 datasets: Toward ontology learning with large language models," *Open Conference Proceedings*, vol. 4, Oct. 2024.