LLM-based Guided Generation of Ontology Term Definitions

Stefan Bischof[0000−0001−9521−8907], Erwin Filtz[0000−0003−3445−0504], Josiane Xavier Parreira[0000−0002−3050−159X], and Simon Steyskal[0000−0002−5183−2486]

Siemens AG Österreich

Abstract. This paper describes our approach for leveraging LLMs to generate definitions and descriptions for ontology terms. Our approach is grounded in the need for detailed and accurate representation of (domain-specific) Knowledge Graphs, and it aims at speeding up the process of generating such text. We outline our approach, including the problems that we encountered, and the solution we propose to overcome them. Our approach is currently in use in an industrial setting.

Keywords: Ontology Engineering · Large Language Models · Text Generation

1 Introduction

Knowledge graphs currently experience an increased uptake by industries. Different companies are turning to ontologies and Knowledge Graphs to enable interoperability within their businesses. In this modelling process, providing accurate term definitions and thorough descriptions of terms in an ontology is crucial, as they support users in having a common understanding of the underlying schema.

Writing such terms' definitions is, however, a labour-intensive task, requiring extensive manual labour to check domain literature and standards for accuracy. This process is also not only time-consuming but also prone to errors and inconsistencies, due to the subjective interpretation of the literature. Therefore, ontology terms will often lack a proper description.

The rise of Large Language Models (LLMs) in the past years has heavily influenced research in multiple domains. LLMs offer powerful features, such as the means to automate certain tasks, therefore saving time and effort of users and developers. In particular, a number of code libraries or library extensions have been proposed to support ontology engineering with the use of LLMs. OntoGPT [1] is a Python library capable of extracting entities and their relationships from natural text and transforming them into another structured format, for instance OWL. Similarly, a Protege plugin [2] converts natural language into OWL. Furthermore, there are tools available (e.g., [3,4]), which take advantage of LLMs for automatically populating a Knowledge Graph, by extracting the entities from text documents, given an existing ontology.

While LLMs provide a promising solution to the tedious manual labour involved in ontology engineering, they are not without their own set of challenges.

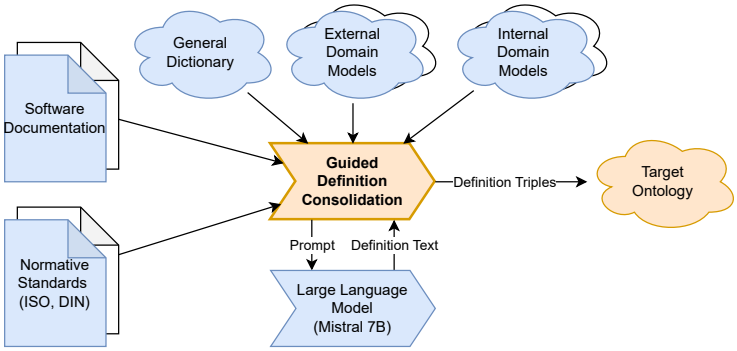

Fig. 1. Architecture of our proposed solution with different types of authoritative input, the main component, the LLM and the target ontology.

The quality of the output they generate is often inconsistent and unpredictable. Instances of "hallucinations", or the generation of inaccurate or irrelevant information are common. These issues not only undermine the accuracy and reliability of the generated term definitions, but also complicate the process of ontology development, as significant time and resources must be then spent on reviewing and correcting these errors. Consequently, these challenges, if not addressed properly, might negate the benefits of using LLMs in the first place. Therefore, a more efficient and reliable approach is needed.

In this paper we propose leveraging LLMs to generate text for definitions for ontological terms, to significantly expedite the process and decrease the time domain experts need to complete such tasks. The approach was developed and tested and it is currently being used in an industrial setting.

In the following section we will outline our approach, which involves a combination of authoritative input and guided text generation.

2 Proposed Solution

In order to take advantage of LLMs and avoid their pitfalls, we propose a solution which guides LLMs by feeding them authoritative input on the terms, similar to existing retrieval-augmented generation (RAG) approaches. This approach aims to steer the generation process, reducing hallucinations and improving the overall quality of the output. An overview of our approach is shown in Fig. 1.

Authoritative Input Authoritative input refers to credible and reliable information sources on the terms. This input serves as a reference guide for the LLMs, helping them generate accurate and relevant content. By providing the LLMs with a concrete foundation, we can reduce the likelihood of hallucinations and improve the consistency of the generated content. The authoritative input sources can be classified into the following groups: (*i*) External dictionaries, providing

Listing 1. Sample Prompt Template

```
You are a domain expert for %domain% tasked with providing an
insightful definition of specific concepts in the context of %context%.

Rules to be followed while generating the definition:
- Desired Length: Short: 1 sentence
- Format: Single sentence
-----------------------------------------------
Generate Definition for:
- Concept Name: %label%
- Definition to be generated:
  As a domain expert, Provide a definition for the concept '%label%'.
  Use the following existing definitions as a basis for yours.
- Existing Definitions:
   * %existing definition 1%
   * %existing definition 2%
   * ...
```

descriptions for terms; (*ii*) External domain models, containing descriptions of terms, such as ASHRAE Standard 223P[1], Brick[2], or Project Haystack[3]; (*iii*) Internal domain models, developed within Siemens; (*iv*) Software documentation; and (*v*) Normative standards, containing a section defining the used terms in the respective standards.

Guided Generation Guided generation involves using the authoritative input as a guide or reference for the LLMs. Instead of generating content from scratch, the LLMs use the input to inform their generation process. This approach ensures that the generated content aligns with the authoritative input, thereby improving the accuracy and reliability of the definitions. Furthermore, by limiting the scope of the generation process to the parameters defined by the authoritative input, we can significantly reduce the likelihood of hallucinations and improve the overall efficiency of the process. A sample prompt for the generation of a description for a term, setting the scene for the LLM, and including definitions taken from the authoritative input is shown in Listing 1. Algorithm 1 describes the ontology annotation process using our proposed approach. After loading the data, for each ontology concept, definitions are searched in the data sources. In case no definition is found, the LLM is also asked for a definition. After that, the returned definitions are used as an input for the prompt generation by the LLM. Finally, a consolidated definition of the returned results is generated by the LLM.

Application and Benefits We have implemented our approach and tested it within a project in our company, which involves creating an ontology for a do-

[1] https://explore.open223.info/
[2] https://brickschema.org/
[3] https://project-haystack.org/

Algorithm 1 Ontology Annotation Process

Input:
- `data_sources`: List of data sources
- `llm_models`: List of Language Learning Models (LLMs)
- `external_definitions_url`: URL for external definition source
- `output_file`: Name of the output file

Output: annotated concepts

Abstract Steps:

1. Load data sources, and build indices.
2. Query indices for information, fetch external definitions.
3. **for** each concept in ontology **do**
4. Query external definitions (fallback to LLM if needed).
5. Create prompts and use LLM to generate definitions.
6. Condense generated definitions using LLM (fallback if needed).
7. **end for**

main in the smart building area. We have observed that our solution produces very good suggestions for concept definitions in the ontology. As a last step an expert reviews the generated definitions to ensure their soundness and correctness. First tests show considerably less time is required for this reviewing process compared to creating the descriptions manually in the first place. A thorough evaluation will be conducted in future work.

3 Conclusions

This paper addressed the task of generating descriptions for terms in an ontology. We have presented our approach which leverages LLMs to support domain experts in executing this task more efficiently. Our approach makes use of authoritative input to guide the text generation process, and it is currently being used in a industrial setting with clear benefits.

References

1. Caufield, J.H., Hegde, H., Emonet, V., Harris, N.L., Joachimiak, M.P., Matentzoglu, N., Kim, H., Moxon, S., Reese, J.T., Haendel, M.A., Robinson, P.N., Mungall, C.J.: Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): a method for populating knowledge bases using zero-shot learning. Bioinformatics **40**(3) (02 2024). https://doi.org/10.1093/bioinformatics/btae104
2. Mateiu, P., Groza, A.: Ontology engineering with large language models. CoRR **abs/2307.16699** (2023). https://doi.org/10.48550/ARXIV.2307.16699
3. Mihindukulasooriya, N., Tiwari, S., Enguix, C.F., Lata, K.: Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In: ISWC'23. LNCS, vol. 14266, pp. 247–265 (2023). https://doi.org/10.1007/978-3-031-47243-5_14
4. Yu, S., Huang, T., Liu, M., Wang, Z.: BEAR: revolutionizing service domain knowledge graph construction with LLM. In: ICSOC'23. LNCS, vol. 14419, pp. 339–346 (2023). https://doi.org/10.1007/978-3-031-48421-6_23