Towards LLM-augmented Creation of Semantic Models for Dataspaces

Sayed Hoseini[1], Andreas Burgdorf[2], Alexander Paulus[2], Tobias Meisen[2], Christoph Quix[1,3] and André Pomp[2]

[1]*Hochschule Niederrhein, Krefeld, Germany*
[2]*Institute for Technologies and Management of Digital Transformation, University of Wuppertal, Wuppertal, Germany*
[3]*Fraunhofer FIT, St. Augustin, Germany*

Abstract

Dataspaces aim to enable smooth and reliable data exchange between different organizations. They have gained increasing attention in Europe following the enactment of the European Data Governance Act. This legislation emphasizes trust, accessibility, and shared dataspaces, which require semantic interoperability grounded in the FAIR principles. Although semantic descriptions in the form of semantic models and ontologies are integral to dataspaces, their full potential remains underutilized. Meaningful metadata, including contextual information, enhances data usability, but manually creating semantic models can be challenging. Large Language Models (LLMs) offer a new way to utilize data in dataspaces. Their advanced natural language processing capabilities enable context-aware data processing and semantic understanding. This paper presents initial experiments on customizing and optimizing LLMs for semantic labeling and modeling tasks. The contributions of this work include research questions for future investigations, early experiments demonstrating the applicability of LLM for semantic labeling, and proposed directions to address discovered challenges.

Keywords

Dataspace, Semantic Modeling, LLMs

1. Introduction

The European Data Governance Act [1, 2] outlines definitions and objectives aimed at bolstering trust, broadening data accessibility, and promoting shared dataspaces. Its impact extends across various data consumers and providers from academia as well as businesses. Efficient data sharing within dataspaces necessitates semantic interoperability as an essential design principle, grounded in the required adherence to FAIR principles [3]. Thereby, the utilization of semantic descriptions and ontologies are already part of many dataspaces, but the potential is far from being fully utilized in actual implementations [4]. An example of this is semantic interoperability and integration, a key aspect of dataspaces that requires aggregating and integrating large

The Second International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, May 26 – 27, 2024, Hersonissos, Greece

sayed.hoseini@hs-niederrhen.de (S. Hoseini); burgdorf@uni-wuppertal.de (A. Burgdorf); paulus@uni-wuppertal.de (A. Paulus); meisen@uni-wuppertal.de (T. Meisen); christoph.quix@hs-niederrhen.de (C. Quix); pomp@uni-wuppertal.de (A. Pomp)

0000-0002-4489-9025 (S. Hoseini); 0000-0001-7776-8497 (A. Burgdorf); 0000-0002-0774-1528 (A. Paulus); 0000-0002-1969-559X (T. Meisen); 0000-0002-1698-4345 (C. Quix); 0000-0003-0111-1813 (A. Pomp)

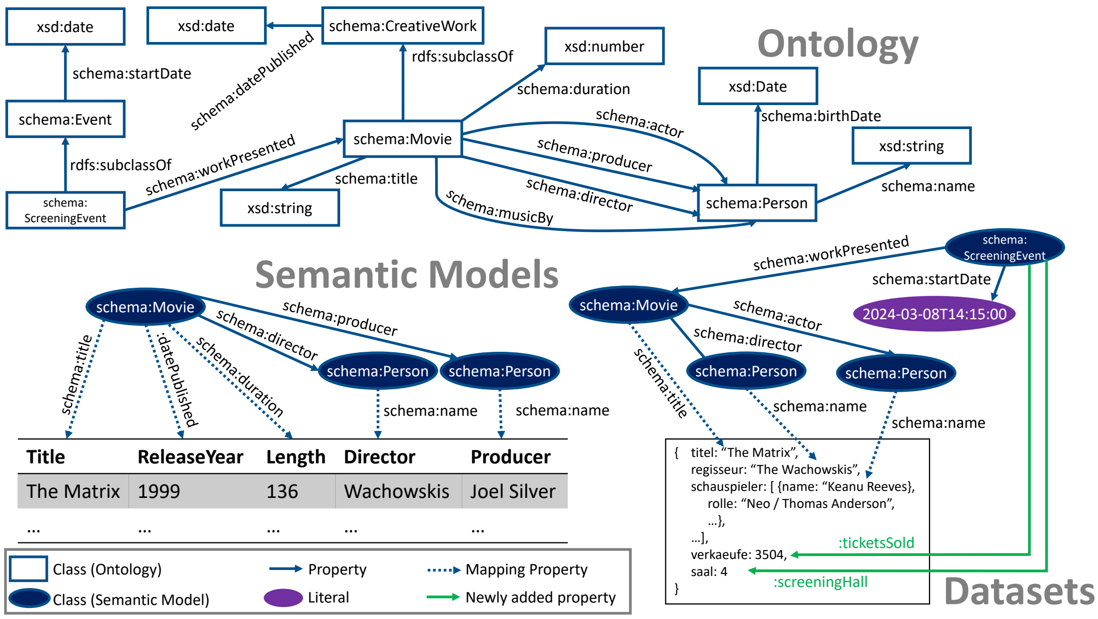
amounts of heterogeneous data from different sources. Managing data can be challenging, not only due to the variety of data formats such as XML, CSV, JSON, relational data, and graph data. In addition, data is often distributed across different departments within an organization, under different governance regimes, and data models. It is important to have a clear and logical structure of information, which fosters a common understanding in dataspaces, i.e., *a lingua franca for data moderation* [5] based on the Linked Data principles.

Meaningful metadata is crucial for enhancing data usability, particularly for users with limited domain knowledge or those unfamiliar with a dataset. Annotating raw data from heterogeneous data sources with semantically rich models enhances data interpretability and usability [6, 7]. This type of semantic data expands beyond typical extractable metadata, such as schema, data types, sizes, and formats, to include contextual information that is not inherent to the specific data source. The field of *Semantic Data Management* (SDM) [8] aims to represent the metadata about heterogeneous data sources in the form of ontologies or knowledge graphs (KG) serialized in a language of the Semantic Web. Hence, the goal is to establish an additional layer between the data and the knowledge layer [9]. This is highly relevant for dataspaces because they integrate data from various systems and platforms, which requires data to be interoperable and seamlessly exchangeable between systems. In order to implement SDM in practice, conceptualizations in the form of KGs and/or ontologies [10], and a mapping between concepts and data items are required. *Semantic models* provide these mappings from single datasets to a common data model to represent data consistently across different applications in a way that is understandable and interpretable by both humans and machines [11].

With companies increasingly acknowledging the importance of data for their business operations, semantic descriptions are often integrated into data management and governance strategies [12, 13], where an ontology or KG serves as a conceptual representation of an organization's data assets. A data source that is semantically well-annotated can be identified and interpreted by leveraging conceptual representations of the data and by comprehending the provided context information stored in the model. However, a huge initial overhead, coming from the time-consuming manual process of creating meaningful semantic descriptions for data sources, hampers the widespread adoption of SDM in practise [8]. Creating semantic models entails deciphering the existing data source, consulting appropriate conceptualizations, and establishing connections between data attributes and concepts provided by the conceptualization.

Automating this task can be challenging and complex. Futia et al. [14] present a method based on graph neural networks covering the process of the process of semantic modeling. However, the model can only optimize semantic models for which historic training data exists. Xu et al. [15] train a cross-modal network to learn semantic features between data sources and semantic models. They admit that the method has shortcomings in dynamically augmenting the semantic models to cover concepts that are not part of the original training data. Moreover, challenges remain with detecting and correcting potentially incorrect attribute types if a source attribute has more than one attribute type, and distinguishing similar attributes with the same entities and semantic types.

Following the rise of Large Language Models (LLMs), one can expect to see a major impact on the landscape of data utilization and exchange within dataspaces. LLMs, such as OpenAI's *GPT-3.5* and *GPT-4.0*, have demonstrated remarkable capabilities in understanding, generating, and processing vast amounts of textual data [16, 17]. Their abilities in natural language processing



Title	ReleaseYear	Length	Director	Producer
The Matrix	1999	136	Wachowskis	Joel Silver
...	...	...	...	...

```
{   titel: “The Matrix”,
    regisseur: “The Wachowskis”,
    schauspieler: [ {name: “Keanu Reeves},
        rolle: “Neo / Thomas Anderson”,
        ...},
    ...],
    verkaeufe: 3504,
    saal: 4
}
```

Figure 1: Semantic models creating a unified mapping from different data sources (datasets) to an ontology.

enable advanced semantic understanding and context-aware data processing within dataspaces. A promising field of LLM application is the integration of heterogeneous data sources stored in a dataspace.

In this article, we highlight some initial experiments in this direction to examine the question of how such general-purpose AI systems can be customized and optimized for data integration tasks in the sense of SDM. In particular, we make the following contributions:

- A set of research questions to be answered in future research endeavors
- Early experiments to illustrate the applicability of LLMs to the tasks of semantic labeling
- Potential future research directions to address the identified challenges with the applicability of LLMs to the tasks of semantic labeling and modeling.

2. Semantic Data Management

Figure 1 illustrates the basic idea of semantic models. The raw datasets in the dataspace are represented at the bottom; they can be in different formats and structures, such as tabular data or hierarchical JSON data, but have partially overlapping content. The semantic model is a *projection* from a shared conceptualization onto the different datasets. It utilizes relevant entities and relationships of the conceptualization, in this case, the *schema.org* ontology (prefix *schema:*), to formalize the context information of the dataset. An essential part of each semantic model are the mappings, indicated as dotted lines, which link attributes in the datasets to classes in the semantic model using properties of these classes. These elementary mappings are referred

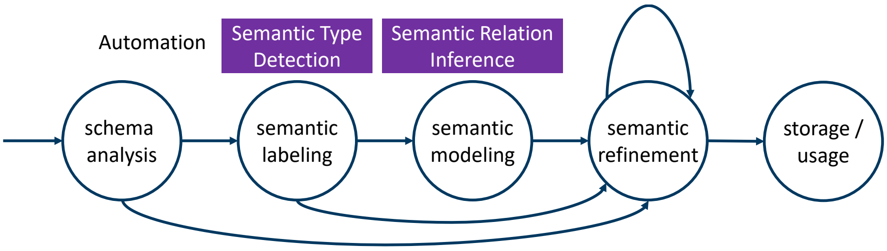


Figure 2: The semantic model creation process as formalized by [21]. Specific terms used for available automation are stated in purple.

to as *semantic labels*. The semantic model captures the precise meaning of the dataset, explicitly encoding the semantic types and relationships among its attributes within the graph.

Following the definition of semantic labeling by Pham et al. [18], a semantic label is an annotation of a dataset attribute by a tuple consisting of a class (subject) and a property (predicate). In this work, a semantic label is represented as a triple: (subject, predicate, schema attribute). For example, the semantic label of the table's column 'Title' is constructed through the subject 'schema:Movie' and the predicate 'schema:title' modeling the relationship between them. This connects the table's content to the attribute 'titel' in the JSON object, indicating an entry point for data integration between the two (heterogeneous) datasets. Moreover, the semantic model doesn't merely rely on a static conceptualization; it can also introduce novel classes and properties [19]. The necessity for evolution becomes apparent when users contribute datasets containing concepts and relationships not yet included in the conceptualization. The semantic label for the JSON key 'verkaeufe' is represented as the triple: (schema:ScreeningEvent, :ticketsSold, 'verkaeufe'). Here, the predicate is a novel property for that specific domain, which is not (yet) present in this form in the general-purpose schema.org ontology. This new knowledge can be systematically integrated, thus perpetually advancing the conceptualization layer [20]. Semantic models complement extractable metadata (such as data types, sizes, formats, etc.) to convey context information that may not be inherent to the dataset at hand, for instance, a starting date of a 'schema:ScreeningEvent' as shown in Figure 1.

The underlying process of semantic model creation has been formalized by [21] and is visualized in Figure 2, starting with the identification of the schema of the dataset, followed by a *semantic labeling* phase, in which basic concepts are assigned to the identified attributes. Automated semantic labeling, referred to as *Semantic Type Detection* [22], is the process of identifying these labels using algorithms and machine learning models. Subsequent to the semantic labeling, the *semantic modeling* phase builds the remaining semantic model by formalizing the context information. During semantic modeling, *semantic relation inference* [14] refers to the process of automated identification of relationships and additional concepts, resulting in a generated semantic model. All automation is followed by the semantic refinement phase, where the modeler is involved in the modeling process to correct any errors present before the semantic model is finalized and stored for documentation purposes. In practice, semantic relation inference depends heavily on accurate semantic labels [23, 24], which underscores the importance of semantic type detection in fully automated systems to induce as few errors as possible for the modeler to correct.

2.1. Research Questions

With an introduction to the field of SDM at hand, we move on to formulate the research questions that have motivated this work. The goal is to leverage LLMs to improve the automation of semantic model creation for large quantities of heterogeneous data sources that share a common domain in a dataspace.

- **RQ1:** How to utilize LLMs to perform semantic type detection with a fixed set of labels coming from a pre-selected conceptualization (such as *WikiData*, or *schema.org*)?
- **RQ2:** How to utilize LLMs to perform semantic type detection against an arbitrary domain ontology, i.e., with no labeled dataset or zero-shot classification?
- **RQ3:** How can LLMs be utilized to identify and formalize the context of a given dataset, creating a full semantic model?

3. Related Work

First, in the scope of this contribution, we consider only works after the launch of *ChatGPT* in November 2022. While one can make a strong case that (large) language models have existed before this date, we decided to draw a line due to the massive performance increase which was quite suddenly accessible to the public. Most of the works found in this limited range are pre-prints that have not yet been peer-reviewed and published in scientific journals. To the best of our knowledge, so far, except for the below-mentioned approaches, there seems to be no further LLM-based efforts on the integration of the semantics of several heterogeneous data sources modeled directly in a language of the Semantic Web in order to generate semantic models in the sense of Figure 1.

Korini et al. [25] are among the first to report the application of LLMs for the Column Type Annotation (CTA) task. CTA is a schema-level annotation task that represents a simplified form of our interpretation of semantic labeling (no predicate) as it aims to map the underlying table schema to a conceptualization. They view CTA as a multi-class classification problem and evaluate different prompt designs. One important artifact that is highlighted is that *ChatGPT* tends to ignore the instruction to use terms from the label space, and instead answers using different terms. This is a known drawback of contemporary LLMs known as the Hallucination problem [26]. Their proposed solutions for this challenge involve first determining a set of classes of the entities described in the table and depending on this set asking *ChatGPT* to annotate columns using only the relevant subset of the overall vocabulary. The evaluation of the approach reports competitive performance when evaluated against the more traditional models which are mostly directly fine-tuned for the CTA task and require significant amounts of task-specific training data [27].

A dataspace stores heterogeneous data of any type of which the majority may be in some form relational, but an important fraction may be more complex, e.g. nested and even unstructured (video, text, audio, ...). We found several works [28, 29, 30] that aim at customizing LLMs via fine-tuning for tables in particular. The goal is to solve the challenge of table understanding which is closely related to understanding the semantics of a data source as it includes the CTA task for example. Usmani et al. [13] highlight the importance of multi-modal knowledge graphs for dataspaces, present a review of the current state, and propose an ontology towards further development. Furthermore, since important use cases for datasets can be attributed to numerical data, it is important to have solid numerical reasoning skills [31]. Several works suggest that modern LLMs excel in simple problem settings, but they fall short of human expert performance in problems requiring numerical reasoning over long contexts. As the complexity of challenging mathematical problems increases, LLMs currently exhibit suboptimal performance [17, 32].

There exist several works that investigate the use of LLMs for Knowledge Graph Engineering [33, 34, 35]. Here the goal is to utilize the LLM for common tasks related to KGs. For example, Meyer et al. [36] investigate SPARQL query generation as well as knowledge extraction from fact sheets and KG exploration among others. They present several prompts that essentially test how to pose questions in natural language executed against serialized KGs. The experiments show that LLMs can return syntactically correct SPARQL queries and even entire serialized RDF models with the desired form based on a task formulated in natural language. Further, LLMs can find relationships in KG and answer basic questions correctly, e.g., *"Are there any connections between US and UK?"*. They report major performance differences between *GPT-3.5* and *GPT-4* in favor of the latter. One particular prompt aims at extracting knowledge from tables and converting it into a serialized KG, which is very close to the idea of semantic modeling. The experiment illustrates several problems with the output of contemporary LLMs:

- A tendency to prioritize the usage of *schema.org* vocabulary. While this works well for well-known entities and properties, the LLMs invent reasonable, but non-existent classes and properties (in the *schema.org* namespace) for concepts and relations that are too specific.
- Non-deterministic output: For multiple runs of the same prompt to the LLM, the output varies. For instance, while in three out of four runs a printer manufacturer was represented as a separate typed entity, in one run it was only expressed as a string literal.
- Invention of non-existent properties, prefixes, and classes: If the LLM cannot identify a fully matching class for a concept or a relation, URIs for those elements are invented for the raw RDF output. While this would be possible in its own namespace, the classes and properties are placed in existing namespaces, such as *schema.org*, resulting in the generated URIs not being resolvable.
- Non-functional queries: SPARQL queries generated by *ChatGPT*-3 did not return the expected results when executed against a knowledge graph, albeit being syntactically correct. All queries needed slight modifications to work, such as correcting the referencing of non-existent classes.

To conclude, the results obtained by Meyer et al. show that the problems commonly observed with LLMs also limit their ability to conduct tasks in the semantic domain. It is therefore not possible to use LLMs out-of-the-box for semantic relation inference. Since semantic type detection is simpler than semantic relation inference and also relies heavily on obtaining context, this area of automation is investigated more closely.

4. Semantic Type Detection with LLMs

To investigate the suitability of LLMs for semantic type detection, we conducted four exemplary experiments using *ChatGPT* 4.0. Therefore, we manually selected three datasets from the VC-SLAM corpus [37] which contains datasets created by human modelers in combination with their data description and semantic model as evaluation datasets. Dataset 1 (VC-SLAM 0001) has seven labels that are close to natural language, Dataset 2 (VC-SLAM 0018) has 21 labels that are mostly human readable, and Dataset 3 (VC-SLAM 0068) consists of 24 labels, some of which are abbreviations. The experiments are briefly described in the following and the results are given in Table 1.

4.1. Experiments

Experiment 1 - Mapping to VC-SLAM: This experiment explores *ChatGPT*'s ability to map dataset labels to the corresponding concepts within the VC-SLAM ontology, provided solely in Turtle (TTL) format, without any additional contextual information. This setup aims to assess the base capability of *ChatGPT* to utilize the ontology's structure and content for semantic type detection. The task involves presenting dataset labels to *ChatGPT* and instructing it to identify the most fitting ontology concept for each label.

Prompt Experiment 1

- You are a tool for semantic type detection. I will provide you an owl ontology that consists of all the concepts you know. This ontology is called VC-SLAM. Later I will additionally provide the labels of three data sets. For each label you return the fitting concept from the ontology.
- The first data set consists of the following labels: type, longitude, address, latitude, tvm_identifier, pay_by_credit_card, pay_by_cash
  Please return the results in the following form: label,concept

Experiment 2 - VC-SLAM with Documentation: In this experiment, the methodology is similar to the *Mapping to VC-SLAM* experiment, but it includes comprehensive documentation of the VC-SLAM ontology and datasets. This tests the hypothesis that additional contextual information enhances the accuracy of semantic type detection.

Experiment 3 - schema.org Ontology: This experiment shifts the focus to a general-purpose ontology to compare *ChatGPT*'s adaptability and performance with a different ontology structure. This experiment provides insights into the model's versatility and the challenges of applying a broad ontology like *schema.org* to a specific dataset, highlighting differences in specificity and applicability.

Experiment 4 - Simplified VC-SLAM: The final experiment aims to investigate the impact of ontology complexity on semantic type detection accuracy. By reducing the VC-SLAM ontology to only include concept names and their descriptions without further relations, this experiment seeks to determine whether a simplified ontology framework would enhance *ChatGPT*'s mapping accuracy due to decreased complexity and ambiguity.

		dataset 1	dataset 2	dataset 3
	Nr of labels	7	21	24
Total Hits	Mapping to VC-SLAM	4	7	3
	VC-SLAM with Documentation	5	13	5
	schema.org Ontology	7	16	11
	Simplified VC-SLAM	4	9	12
Accuracy	Mapping to VC-SLAM	0.57142	0.33333	0.125
	VC-SLAM with Documentation	0.71428	0.61904	0.20833
	schema.org Ontology	1	0.76190	0.45833
	Simplified VC-SLAM	0.57142	0.42857	0.5

Table 1
The table demonstrates the results of the four experiments that we run on three datasets. The datasets refer to the following original datasets from VC-SLAM [37]: 1: 0001, 2: 0018, 3: 0068

4.2. Results

The outcomes of the experiments are measured across the three different datasets. **Experiment 1 - Mapping to VC-SLAM** reveals a varying performance with a 57.1% accuracy rate for the first dataset (4 out of 7 labels correctly mapped), 33.3% for the second (7 out of 21), and a notably lower 12.5% for the third (3 out of 24). These results indicate that while *ChatGPT* can achieve some level of correct mapping based on the ontology structure alone.

Experiment 2 - VC-SLAM with Documentation showed improved performance with a 71.4% accuracy rate for the first dataset (5 out of 7 labels correctly mapped), 61.9% for the second (13 out of 21), and 20.8% for the third (5 out of 24). These results highlight the significant impact of extra contextual information in enhancing *ChatGPT*'s semantic type detection capabilities, leading to more accurate mappings.

Experiment 3 - schema.org Ontology further demonstrated *ChatGPT*'s adaptability with impressive accuracies: 100% for the first dataset (7 out of 7 labels correctly mapped), 76.2% for the second (16 out of 21), and 45.8% for the third (11 out of 24). Reasons for this may be that ChatGPT is better at handling ontologies that were already part of the training data, or that the descriptions in schema.org are more meaningful than those of the VC-SLAM ontology.

Experiment 4 - Simplified VC-SLAM yielded mixed results: 57.1% accuracy for the first dataset (4 out of 7 labels correctly mapped), 42.9% for the second (9 out of 21), and 50% for the third (12 out of 24). These outcomes suggest that simplification of the ontology does not necessarily lead to improved performance across all datasets, reflecting the complex balance between ontology complexity and the effectiveness of semantic type detection with *ChatGPT*.

During these experiments, several key findings emerged. First, the availability of additional contexts, such as ontology documentation, significantly improves *ChatGPT*'s ability to accurately map dataset labels to ontology concepts, underscoring the importance of rich contextual information for semantic type detection tasks. Second, the experiments revealed *ChatGPT*'s adaptability to different ontologies, with performance variations highlighting the model's capability to handle both specialized and general-purpose ontologies. Lastly, the simplification of the ontology structure was shown to potentially enhance semantic type detection accuracy, suggesting that the complexity of an ontology can affect the efficiency and effectiveness of label mapping. These findings contribute valuable insights into the potential of leveraging LLMs for semantic type detection, indicating promising pathways for automating and refining the data categorization process. The experiments underscore the significance of ontology design and contextual information in optimizing the performance of semantic type detection tasks using AI models like *ChatGPT*.

5. Semantic Model Creation with LLMs

Adding LLMs into the process of semantic model creation provides automation algorithms with the ability to profit from the advantages that these pre-trained models provide. Taking the findings from both related work and our experiments into account, it can be deduced that results obtained from LLMs need to be verified and checked before being applied in a semantic model creation scenario within dataspaces. In the following, two approaches on how to utilize LLM-generated output in automated semantic model creation are conceptualized.

5.1. Unifying KGs with LLMs for Semantic Modeling

LLM		**KG**	
Pros +	**Cons -**	**Pros +**	**Cons -**
General knowledge	Implicit knowledge	Structural knowledge	Incompleteness
Language processing	Hallucinations	Accuracy	Lacking language understanding
Generalizability	Indecisiveness	Decisiveness	
	Black-Box	Interpretability	Unseen facts
	Lacking domain-specific/ new knowledge	Domain-specific knowledge	
		Evolving knowledge	

Table 2
The pros and cons of LLMs vs KGs as described by [38]

Although the technologies for linked data and the Semantic Web have become more mature in recent years, the amount of data considered in Semantic Web applications is far less than in Big Data applications [39]. Thus, scalability to large, heterogeneous data sets is a major challenge for applying Semantic Web technologies in dataspaces for which LLMs can be a great help. However, even though LLMs can effectively possess rich knowledge learned from massive amounts of training data and benefit downstream tasks at the fine-tuning stage, as previously described, they still have significant limitations due to the lack of external knowledge [17]. In contrast, KGs are structured knowledge models that explicitly store rich factual knowledge. However, KGs are difficult to construct and evolve by nature, making it challenging to generate new facts and represent unseen knowledge [40]. Therefore, it is reasonable to view KGs and LLMs as two complementary technologies whose integration has the potential to produce synergy, capitalizing on the strengths of each while mitigating their respective weaknesses.

For a detailed discussion on research towards the unification of language models and KGs we refer to the survey by Pan et al. [38]. They contrast the pros and cons of (large) language models vs KGs. Table 2 confirms the previous findings about the drawbacks of LLMs, namely

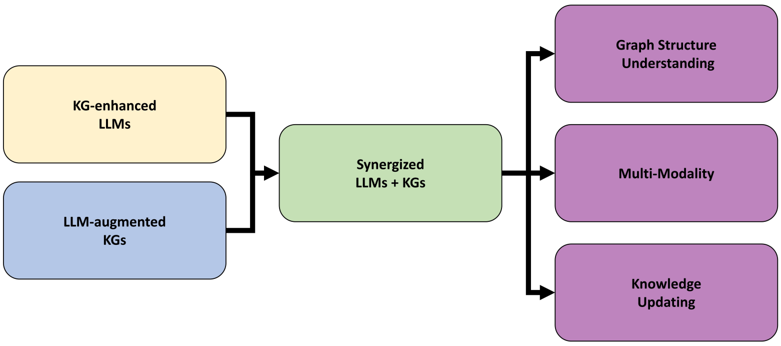


Figure 3: The unification of KGs and LLMs. Adopted from [38].

hallucination, a lack of domain-specific knowledge, indecisiveness, and a lack of interpretability. Conversely, the automated construction of KGs is equally challenging, and current approaches to KGs are inadequate in handling the incomplete and dynamically changing nature of real-world KGs. Additionally, many of the current techniques for KGs are tailored to particular tasks and, therefore not easy to generalize to broader applications. This suggests that KG and LLMs indeed complement and may synergize with each other. Pan et al. further predict three main directions for future research toward this goal: **KG-enhanced LLMs**, which incorporate KGs during the pre-training and inference phases of LLMs to enhance understanding of the knowledge learned by LLMs. Here we direct the interested reader to the survey by Hu et al. [41]. Then there are **LLM-augmented KGs**, mentioned in a similar form by Meyer et al. [36] (see Section 3). Ultimately, these directions may be integrated to produce **Synergized LLMs + KGs**, in which LLMs and KGs play equal roles and work in a mutually beneficial way to facilitate reasoning driven by both data and knowledge. This fusion may possibly address some of the contemporary challenges discussed in Section 3 and represent one answer to the third research question **(RQ3)** on how LLMs can enhance the semantic model creation process. Figure 3 illustrates the integration of the two directions and its stated goals. As the most basic semantic unit, entities play a crucial role, and incorporating their knowledge into LLMs helps to improve semantic understanding. In addition, there are also a large number of relational triples in the knowledge graph, which can provide sufficient structured information to further improve the semantic understanding. Since conventional LLMs trained on plain text data are not designed to understand (graph-)structured data such as knowledge graphs, they might not fully grasp or understand the information conveyed by the KG structure. This assumption is confirmed by our experiments (see Section 4), since reducing the representation of ontologies to plain text significantly improves the performance. This indicates that *ChatGPT* does not handle the graph structure well. Synergized LLMs + KGs promise to be able to understand the underlying graph structure which could improve the performance of KG technology e.g. in discovering unseen facts and exploration for example.

Multimodal KGs are becoming increasingly important for dataspaces as they integrate different modalities, including text, image, audio, and video data, into a single graph, allowing for a comprehensive representation of complex data [13]. As previously described, it is important for semantic modeling to have solid numerical reasoning skills. Similar to KGs, regular datasets

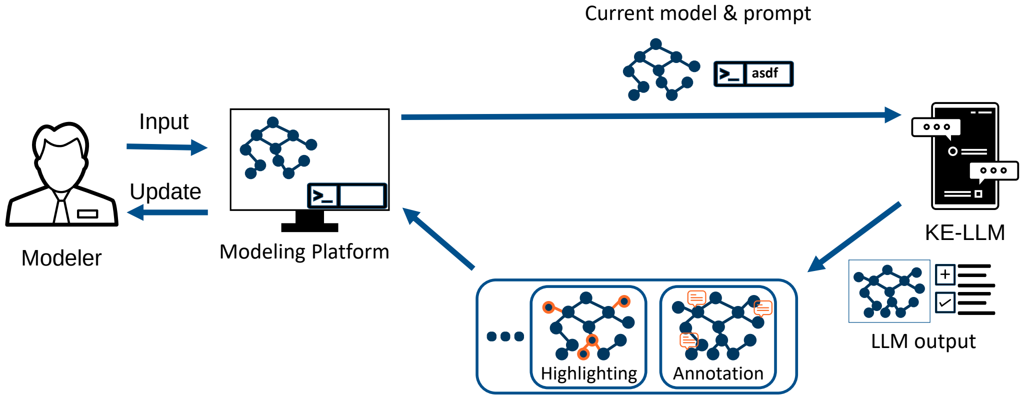


Figure 4: Communication-based interactive semantic model generation process

like CSVs or JSONs can be viewed as (semi-)structured data that represent a further modality. Therefore, effectively leveraging representations from multiple modalities, in particular tables and spreadsheets, would be a significant milestone towards the unification of KGs and LLMs.

Finally to remedy the problems with hallucinations and updates of the internal knowledge of LLMs as real-world situations change, the incorporation of knowledge from KGs represents a logical solution. For hallucination, the KGs can be leveraged as an external source to validate or fact-check the output of an LLM. Editing the knowledge of an LLM live without re-training is an attractive idea. However, current methods have severe problems, and further research is required [42, 43]. A potential solution to this problem is presented in the next section.

5.2. LLM-Supported Interactive Semantic Model Design

Since semantic model creation is usually performed inside a semantic modeling platform, integrating LLMs into the semantic model creation following an interactive pattern requires the surrounding platforms to offer additional functions. None of the existing semantic modeling platforms, such as *SAND* [44], *MantisTable* [45] or *PLASMA* [46], offer the ability to communicate with a generative AI to refine semantic models. Future platforms to support manual semantic model creation will likely integrate the interaction with an LLM as a central component of their design. For example, a semantic model creation could be fitted into a session with an interactive LLM. Any type of generative LLM can be used, however, using knowledge-enhanced LLMs (see Section 5.1) helps to reduce the effects of unwanted phenomena, such as hallucinations.

Users input their desired changes using natural language, and the LLM alters the model accordingly. This process requires two central features to be realized: First, users must be able to formulate their changes using a prompt-like interface. Any identified shortcomings can be expressed in text form, even using natural language, to interact with the system and improve the usability of the semantic modeling platform for users with little or no previous knowledge of semantic technologies. Additionally, the platform should provide a process for piping and filtering LLM output to minimize the impact of known drawbacks, such as hallucinations.

Figure 4 visualizes this process in which the modeler and the LLM serve as the interacting participants of a communication. All interactions between both parties are conducted through various services, such as the modeling platform and the LLM's API. These services apply modifications and transform the contained data to match the other side's data model. For example, when a semantic model generation is requested using an LLM, the current semantic model is provided to the LLM, preferably using a pre-configured, session-based GPT specialized in semantic model creation. The request is appended to the interactive session, resulting in an updated model being generated by the LLM. The LLM's extensive knowledge and advanced capability to process natural speech input allows it to modify the semantic model based on the modeler's intentions, proposing a formalized solution to shortcomings such as syntactical errors. The LLM-generated output undergoes post-processing to ensure presentability to the modeler, particularly when generating large semantic models. The changes made by the LLM in the last iteration are highlighted in separate steps in the generated model, making it easy to identify the changes made based on the last input when displaying the results in the modeling platform. In case the LLM generates corresponding textual output, it is parsed and attached to the updated model using a special set of RDF properties. This enables the modeler to verify the reasoning behind the modifications made to specific elements. Once the post-processing is complete, the proposed semantic model is transferred back to the user and displayed.

6. Conclusion

This article explores the applicability of modern LLMs for semantic data management in dataspaces, in particular to the tasks of semantic model creation. Our objective is to address the provided research questions to offer directions of future research for preparing LLMs for the complex task of creating semantic models for vast amounts of heterogeneous data sources in dataspaces.

Regarding **RQ1** and **RQ2**, the experiments in Section 4 demonstrate the feasibility of utilizing LLMs for semantic type detection with a fixed or limited set of labels derived from legacy knowledge graphs. LLMs show promise in achieving significant accuracy in semantic type detection tasks, especially when additional contextual information or documentation is provided alongside the ontology. in particular, Experiment 3, which used the *schema.org* ontology, showcases the high adaptability and potential of LLMs to accurately map dataset labels to ontology concepts with an accuracy reaching up to 100% for certain datasets. This indicates that LLMs can serve as a powerful tool for semantic type detection. Experiment 4's approach, using a simplified version of the VC-SLAM ontology, offers insight into how LLMs might tackle semantic type detection tasks when the ontology is minimized to basic concept names and descriptions, achieving up to 57.1% accuracy in some cases. The findings suggest that LLMs, including *ChatGPT*, can effectively engage in semantic type detection tasks even when presented with new, unfamiliar, or arbitrary domain ontologies, by leveraging their inherent understanding of language and context.

Regarding **RQ3**, exploiting the vast knowledge and reasoning capabilities of LLMs to automate semantic modeling is an attractive idea. However, significant research is still necessary to integrate KGs with LLMs to produce synergy between these two complementary technologies (see Section 5.1). LLMs do not navigate on graphs or handle numerical data sets well. They may suffer from hallucinations and cannot acquire domain-specific knowledge [47] easily.

The LLM-supported interactive semantic model design (see Section 5) establishes a unique way of generating semantic models, providing another possible answer to **RQ3** on how LLMs can enhance the semantic model creation process. However, it requires several additions to today's semantic modeling platforms. In theory, the creation of a semantic model can be a fully immersive experience, where modifications can even be made through voice commands. These modifications are then converted to prompts and interpreted by the natural language processing capabilities of LLMs. The resulting changes are automatically visualized, effectively utilizing the LLM as a semantic modeling system. While the presented results and concepts represent a first approach to the topic, the stated research questions remain open to inspire future research in this area.

Acknowledgements

This work has been sponsored by the German Federal Ministry of Education and Research in the funding program "Forschung an Fachhochschulen" (grant no. 13FH557KX0) and funding program "Datenkompetenzzentren für die Wissenschaft" (grant no. 16DKZ2056B).

Declaration of generative AI and AI-assisted technologies in the writing process: During the preparation of this work, the author(s) used OpenAI's generative AI (ChatGPT v3.5 & v4), *DeepL* and *Grammarly* to improve the writing, make suggestions, and for rephrasing. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] J. Baloup, E. Bayamlıoğlu, A. Benmayor, C. Ducuing, L. Dutkiewicz, T. Lalova-Spinks, Y. Miadzvetskaya, B. Peeters, White paper on the data governance act, CiTiP Working Paper 2021 (2021).

[2] L. Nagel, J. J. Hierro, E. Perea, D. Lycklama, C. Mertens, A.-S. Taillandier, M. Marques, J. Gelhaar, A. Marguglio, U. Ahle, et al., Design Principles for Data Spaces: Position Paper, Technical Report, E. ON Energy Research Center, 2021.

[3] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016).

[4] J. Theissen-Lipp, M. Kocher, C. Lange, S. Decker, A. Paulus, A. Pomp, E. Curry, Semantics in dataspaces: Origin and future directions, in: Companion Proceedings of the ACM Web Conference 2023, WWW '23 Companion, ACM, 2023.

[5] S. Auer, Semantic Integration and Interoperability, Springer International Publishing, 2022.

[6] S. Meckler, R. Dorsch, D. Henselmann, A. Harth, The web and linked data as a solid foundation for dataspaces, in: Companion Proceedings of the ACM Web Conference, 2023.

[7] M. Yahya, J. G. Breslin, M. I. Ali, Semantic web and knowledge graphs for industry 4.0, Applied Sciences 11 (2021).

[8] S. Hoseini, J. Theissen-Lipp, C. Quix, Semantic data management in data lakes, arXiv:2310.15373 (2023).

[9] A. Pomp, A. Paulus, A. Kirmse, V. Kraus, T. Meisen, Applying semantics to reduce the time to analytics within complex heterogeneous infrastructures, Technologies 6 (2018).

[10] A. Hogan, et al., Knowledge graphs, ACM Comput. Surv. 54 (2022).

[11] G. Solmaz, F. Cirillo, J. Fürst, T. Jacobs, M. Bauer, E. Kovacs, J. R. Santana, L. Sánchez, Enabling data spaces: existing developments and challenges, in: Proceedings of the 1st International Workshop on Data Economy, DE '22, ACM, 2022.

[12] H. Dibowski, S. Schmid, Y. Svetashova, C. Henson, T. Tran, Using semantic technologies to manage a data lake: Data catalog, provenance and access control, in: Proc. Scalable Semantic Web Knowledge Base Systems Workshop, volume 2757 of *CEUR WS*, 2020.

[13] A. Usmani, M. J. Khan, J. G. Breslin, E. Curry, Towards multimodal knowledge graphs for data spaces, in: Companion Proceedings of the ACM Web Conference 2023, WWW '23 Companion, ACM, 2023.

[14] G. Futia, A. Vetrò, J. C. De Martin, Semi: A semantic modeling machine to build knowledge graphs with graph neural networks, SoftwareX 12 (2020).

[15] R. Xu, W. Mayer, H. Chu, Y. Zhang, H.-Y. Zhang, Y. Wang, Y. Liu, Z. Feng, Automatic semantic modeling of structured data sources with cross-modal retrieval, Pattern Recognition Letters 177 (2024).

[16] H. Liu, R. Ning, Z. Teng, J. Liu, Q. Zhou, Y. Zhang, Evaluating the logical reasoning ability of chatgpt and gpt-4, arXiv:2304.03439 (2023).

[17] Y. Chang, X. Wang, J. Wang, et al., A survey on evaluation of large language models, ACM Trans. Intell. Syst. Technol. (2024). Just Accepted.

[18] M. Pham, S. Alse, C. A. Knoblock, P. Szekely, Semantic Labeling: A Domain-Independent Approach, in: The Semantic Web – ISWC 2016, Springer International Publishing, 2016.

[19] A. Pomp, Bottom-up Knowledge Graph-based Data Management, Berichte aus dem Maschinenbau, Shaker, 2020.

[20] A. Pomp, J. Lipp, T. Meisen, You are missing a concept! enhancing ontology-based data access with evolving ontologies, in: Proc. ICSC, IEEE, 2019.

[21] A. Paulus, A. Burgdorf, A. Pomp, T. Meisen, Recent advances and future challenges of semantic modeling, in: Proc. 15th IEEE ICSC, IEEE, 2021.

[22] M. Hulsebos, K. Hu, M. Bakker, et al., Sherlock: A deep learning approach to semantic data type detection, in: Proceedings of the 25th ACM SIGKDD, ACM, 2019.

[23] B. Vu, C. Knoblock, J. Pujara, Learning Semantic Models of Data Sources Using Probabilistic Graphical Models, in: The World Wide Web Conference, WWW '19, ACM, 2019.

[24] M. Taheriyan, C. A. Knoblock, P. Szekely, J. L. Ambite, Y. Chen, Leveraging Linked Data to Infer Semantic Relations within Structured Sources, in: Proceedings of the 6th International Workshop on Consuming Linked Data (COLD 2015), 2015.

[25] K. Korini, C. Bizer, Column type annotation using chatgpt, arXiv:2306.00745 (2023).

[26] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, ACM Comput. Surv. 55 (2023).

[27] Y. Suhara, J. Li, Y. Li, D. Zhang, c. Demiralp, C. Chen, W.-C. Tan, Annotating columns with pre-trained language models, in: Proceedings of the 2022 International Conference on Management of Data, SIGMOD '22, ACM, 2022.

[28] P. Li, Y. He, D. Yashar, W. Cui, S. Ge, H. Zhang, D. R. Fainman, D. Zhang, S. Chaudhuri, Table-gpt: Table-tuned gpt for diverse table tasks, arXiv:2310.09263 (2023).

[29] S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, D. Sontag, Tabllm: Few-shot classification of tabular data with large language models, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2023.

[30] T. Zhang, X. Yue, Y. Li, H. Sun, Tablellama: Towards open large generalist models for tables, arXiv:2311.09206 (2023).

[31] S. Gottschalk, E. Demidova, Tab2kg: Semantic table interpretation with lightweight semantic profiles, Semantic Web 13 (2022).

[32] Y. Zhao, Y. Long, H. Liu, et al., Docmath-eval: Evaluating numerical reasoning capabilities of llms in understanding long documents with tabular data, arXiv:2311.09805 (2023).

[33] B. P. Allen, L. Stork, P. Groth, Knowledge engineering using large language models, arXiv:2310.00637 (2023).

[34] L.-P. Meyer, J. Frey, K. Junghanns, F. Brei, K. Bulert, S. Gründer-Fahrer, M. Martin, Developing a scalable benchmark for assessing large language models in knowledge graph engineering, arXiv:2308.16622 (2023).

[35] J. Frey, L.-P. Meyer, N. Arndt, F. Brei, K. Bulert, Benchmarking the abilities of large language models for rdf knowledge graph creation and comprehension: How well do llms speak turtle?, arXiv:2309.17122 (2023).

[36] L.-P. Meyer, C. Stadler, J. Frey, et al., Llm-assisted knowledge graph engineering: Experiments with chatgpt, arXiv:2307.06917 (2023).

[37] A. Burgdorf, A. Paulus, A. Pomp, T. Meisen, VC-SLAM - A Handcrafted Data Corpus for the Construction of Semantic Models, Data 7 (2022).

[38] S. Pan, L. Luo, Y. Wang, et al., Unifying large language models and knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering (2024).

[39] A. Haller, A. Polleres, D. Dobriy, N. Ferranti, S. J. Rodríguez Méndez, An analysis of links in wikidata, in: European Semantic Web Conference, Springer, 2022.

[40] E. Iglesias, S. Jozashoori, M.-E. Vidal, Scaling up knowledge graph creation to large and heterogeneous data sources, Journal of Web Semantics 75 (2023).

[41] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, J. Li, A survey of knowledge enhanced pre-trained language models, IEEE Transactions on Knowledge and Data Engineering (2023).

[42] Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, N. Zhang, Editing large language models: Problems, methods, and opportunities, arXiv:2305.13172 (2023).

[43] R. Cohen, E. Biran, O. Yoran, A. Globerson, M. Geva, Evaluating the ripple effects of knowledge editing in language models, arXiv:2307.12976 (2023).

[44] B. Vu, C. A. Knoblock, SAND : A Tool for Creating Semantic Descriptions of Tabular Sources, in: The semantic web, volume 13384 of *LNCS*, Springer, 2022.

[45] R. Avogadro, M. Cremaschi, Mantistable v: A novel and efficient approach to semantic table interpretation., in: SemTab@ ISWC, 2021.

[46] Alexander Paulus, Andreas Burgdorf, Lars Puleikis, Tristan Langer, André Pomp, Tobias Meisen, PLASMA: Platform for Auxiliary Semantic Modeling Approaches, in: International Conference on Enterprise Information Systems, 2021.

[47] N. Kandpal, H. Deng, A. Roberts, E. Wallace, C. Raffel, Large Language Models Struggle to Learn Long-Tail Knowledge, 2022.