An Agentic Framework for Dynamic Ontology Generation

Ali Amini[1,*,†]

[1]Department of Mathematics and Computer Science, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

Abstract

Ontology design and construction in any domain is usually a very human-oriented tast which needs hours of research and design, particularly for operational business data like restaurant menu or in other hospitality-related areas. There are multiple factors like environments, data randomness, inconsistencies in identifying categories, products changes over the time, and different semantic labels are always having direct impact on how scalable and feasible are any analytical and knowledge management tasks in business intelligence (BI). The old ontology engineering methods were defined by manual schema design by domain experts, changing the schema as needed in future, keeping it updated per new data comes (a fully human-centered process) which is a bottleneck for dynamic business environment in which tons of new data may get generated everyday.

In this paper, we are introducing a new system design which is based on AI Agent for all ontology engineering (construction, management, alignment,..) tasks in business intelligence frameworks. With all advancements that Large Language Models (LLMs) received in past few years, they can be considered as an ontology engineering agent which is embedded in the ontology and data management systems of BI environment instead of being a passive assistant. In AI-Agent embedded ontology systems, multiple Ontology-specific tasks such as finding the objects and semantics between them, aligning new coming objects with the current ones and continuously checking for new concepts are being defined for different agents to handle. The main tasks for ontology engineering systems are including detecting the hierarchies in the schema (e.g. "soft drinks" are subclass of "drinks"), identifying the semantic relationship between the entities, schema evolution as new concepts are being added to the data, and keeping the human validation in the loop. We observed that the collaboration between these agents can help defining ontology development task from a static modeling by human to a continuous adaptive task of knowledge management with more controlled human intervention which is more feasible in dynamic BI environments. For this study, we used semi-structured restaurant menu CSV dataset which has name, category, description and other information on restaurant menu items. This dataset has multiple layer of taxonomies and semantic relationship among the objects which makes it very reliant on manual modeling and analysis.

In this work, we apply evaluation in four levels of agentic process which is aligned with business intelligence aspects: (1) fully automated ontology extraction, (2) recursive semantics check and enrichment, (3) adaptive schema evolution, (4) controlled human intervention. Our results were promising and suggested that these agents involvement could accelerate the construction of the semantic layer of our ontology and improve the adaptability of it to include the business data while having human in the process for validation which is essential. In future, we are developing on scalable plugin-based agents to enterprise integration.

Keywords

Large Language Model, Ontology, Knowledge Graph, Ontology Construction, Semantic Extraction Agent

1. Introduction

As businesses are growing the constant generation of the data is also growing, the need to have a semantic structured data in order to ease the analytics, decision support, and market understanding is getting bigger every day across enterprise systems [1]. In the current status of enterprise and operations dataset there is huge acceptance over unstructured, heterogeneous and semantically inconsistent data, including not limited to, datasets of product catalogs, services and their logs, transactional records of sales, and advertisement and marketing [2]. Many inconsistencies in datasets, particularly in BI domain, is causing frequent limit on scalable analytical workflow and consequently making issue in running analyses or doing data interpretation. All these inconsistencies are driving the team to do multiple duplicated efforts in model and knowledge discovery [3].

In any ontology engineering effort the main focus in on giving a structure to the domain knowledge by extracting conceptual entities like classes, relationship between the objects, and finding the hierarchical

ESWC'26: 5th Workshop on LLM-Integrated Knowledge Graph Generation from Text (TEXT2KG), May 10-14, 2026 | Dubrovnik, Croatia

aliamini99@aut.ac.ir (A. Amini)

schema or taxonomy behind the objects. In fact, ontologies can play the semantic layer role in any system including the BI systems. This layer is responsible for giving a standard logic to business concepts and improve the further integration and analytical process which is happening on top of this layer [4]. Although defining an ontology has many benefit for businesses, it is a very labor-intensive task which needs domain expert on each dataset, constant monitoring and maintenance on data; because of this is it very rare for businesses to put the effort in adopting it.

Large Language Models (LLMs) has been under continuous advancement and grow in the past few years which made it a very powerful tool for various text-based dependencies particularly the knowledge engineering. LLMs were the leading point in processing the structures and unstructured data, interpreting it, extracting the entities and semantics, and generating more semantical data and knowledge representation. In recent researches on the semantic web, there were several discussions on how LLM can help this domain in ontology engineering and knowledge graph construction. What we are observing in the recent researches is more using LLMs as passive tool to help in extracting structure and semantics, for example, using prompts to generate schema. However, the passively use of this powerful tool in this application can fail because this miss any evolution and constant changes that may occur in the dataset and accordingly in the ontology of business environment[5].

In this paper we are introducing an agent-oriented framework for ontology management (construction, alignment, evolvement,..) for business intelligence (BI) workflows. We defined different modules in the process of ontology management and instead of considering the Large Language Models (LLMs) as one time ontology construction tool, we embed the models into different modules (agents) which are responsible for a certain task which is a components in the final decision-support structure for analytics. In every ontology construction task, certain components are defined which are shaping the final capabilities of the module. We defined these components in this research as hierarchical schema induction, semantic relationship reasoning, schema gap detection, and human-in-the-loop validation. With this design ontology engineering can transform from a human-centric static task to dynamic and continuous knowledge engineering task.

We used a real-world restaurant dataset (a flat tabular dataset) to explore the feasibility of our idea and do deeper study on our data for BI. This data may look simple from the structure-wise perspective, but is has multiple layer of hierarchies and semantics like cuisine types, food categories, and ingredient in dishes which requires manual model of domain experts (with cuisine knowledge) to model the ontology and discover the semantics in it. In many cases, the domain experts need to be knowledgeable in international cuisines to understand the semantics of them too. In this proposed agentic approach, knowledge and semantics get extracted automatically and the ontology enrichment process is happening constantly with human involvement for validation.

In this study we are trying to build a framework for continuous ontology engineering task (constructions, evolvement,..) for BI systems which are more exposed to unstructured dataset. We are trying to build an agentic system that handles ontology induction, enrichment, schema evolvement in a dynamic way with human-in-the-loop. We explore that how ontology construction tasks can be modularized and how we can orchestrate the the flow among them dynamically for BI systems. These modules handles tasks from ontology construction, semantic extractions (objects, classes, relationships) from business data and observing if the llm-driven ontology agents can help reducing the human-centered tasks while we have all semantic consistencies.

Traditional ontology engineering approaches were powerful and worked for over two decades, however, because of the dynamic nature of business intelligence sectors they are very limited. First and most important issue with traditional approaches is that they need huge effort by human experts to get together and construct the ontology in long sessions which is both a time consuming and also not feasible to do it repeatedly and rapidly by each dataset changes in the dynamic BI environment. Secondly, in traditional approaches, the ontology design task was defined as a static one-time task which defines the classes, relationships, so on. So basically, each changes in categories, attributes, and data semantics needs another effort by domain expert to redo the design. Third, the ontology are being developed outside their context and being used in operational or business context, which makes their integration very difficult with analytics and decision-support applications. In contrast, this work is proposing ontology development effort as an AI-agent oriented frameworks which tries to construct the ontology in a continuous and semi-automated process embedded in the BI knowledge management systems. We are not encouraging any removal of human interaction in this process, in fact, we are proposing an approach which tries to reduce the manual effort of the domain experts and narrows it does to validation and occasional intervention. This system can improve adaptability to new datasets coming, identify new semantics and enables better integration with knowledge graphs and analytics systems.

First, they require significant manual effort from domain experts, making ontology development time-consuming and difficult to scale across rapidly evolving datasets. Second, ontologies are typically designed as static artifacts, requiring repeated redesign when new categories, attributes, or relationships emerge. Third, they are often developed outside operational pipelines, limiting their integration with real-time analytics and decision-support systems. In contrast, this work proposes an agent-oriented framework that repositions ontology construction as a continuous and semi-automated process embedded within BI workflows. Rather than replacing domain experts, the proposed approach aims to reduce manual effort, improve adaptability to evolving data structures, and enable tighter integration between semantic modeling and analytics systems. In this paper, we are proposing a new agent oriented ontology engineering system which addresses the dependencies of manual human intervention and experts efforts and avoid the total ontology redesign which is costly and difficult to maintain in any dynamic business intelligence environment. Our proposed system is agentic-based system which is continuously observing the data and propose changes and can seamlessly integrate with any dynamic BI pipelines. In this approach, the agents are designed to reduce the frequent redesign of the whole ontology while improving the scalability of that in case larger datasets are integrated. We have also a solid evaluation process which look into multiple factors like ontology richness, ontology adaptation to new data, and minimizing the domain expert intervention in the process. This process involved the practical restaurant dataset to show the applicability and effectiveness of the proposed approach.

Specifically, the framework introduces three key advantages over traditional approaches: (1) reduced human intervention through LLM-assisted ontology induction and enrichment, (2) adaptive schema evolution driven by continuous data ingestion, and (3) direct integration with BI pipelines to support real-time semantic analysis.

2. Related Work

2.1. Manual and Semi-Automated Ontology Engineering

In past two decades, ontology engineering was mainly an expert-driven task and hugely dependent on domain experts and ontologist to manually model the schema, classes, relationship, and axioms using tools like Protégé [6]. Traditional approaches are very precise but they are slow, human-dependent and difficult to scale in dynamic environments of BI pipeline because the data and its semantics might change frequently in these environments. In other systems that are semi-automated, experts try to incorporate NLP techniques like tokenization, part-of-speech tagging, named entity recognition, and pattern recognition to extract the objects and then using rules the infer the triples and semantics which is ending in T-box and A-box creation [7]. Both fully-manual or semi-automated approaches need huge efforts on defining the rules and understanding the data which is limiting their solidness and the ability to be generalized.

2.2. Embedding-based Models

With recent advancement in introducing embedding vectors in past decade, mant ontology engineering efforts started focusing on extracting contextual relationship instead of rule-based or pattern-based methods in ontology processing.[8, 9]. These approach are flexible and more powerful in understanding the context on the data and identifying the related objects and semantics among them but they need a huge and large labeled datasets for training. Training the model on these data can also add noises and false positives because these models cannot handle generalization and usually get failed in that.

2.3. AI-Driven Agents and Generative Approaches

Recent works are showing approaches for semi-automated KG construction pipelines using LLMs for various tasks of ontology design and development, ontology alignments and ontology evaluation with more controlled human efforts. These generative models offer powerful semantic analysis and ontology development in a dynamic manner without referring to any rules. It is also accurate to consider that these generative models might include accuracy, consistency and precision issues particularly in BI domains like food and restaurants[10]. In other published works, authors are introducing LLM-based systems (e.g., OntoEKG, SCHEMA-MINERpro) which are powerful in ontology construction and and understanding the hierarchies from unstructured data; however, in large scaled data they can add noised without human supervising them [11, 12]. With huge advancement in defining and applying AI agents, they can be incorporated for various tasks like ontology induction, reasoning and validation in automated pipelines with more control over the context of the domain and data [13]. These agents are showing promising results as they incorporate with other agents to create a multi-agent collaboration. They get more autonomous and can work with memory in their individual task while keeping the harmony and collaborating in the bigger problem to get solved [14].

3. Dataset And Usecase

We used the Restaurant Menu Items Profitability dataset available on Kaggle [1], which contains structured information on restaurant menu items, including:

- RestaurantID: Unique identifier of the restaurant (e.g., R001, R003).
- MenuCategory: High-level category (e.g., Appetizers, Beverages, Desserts, Main Course).
- MenuItem: Name of the menu item (e.g., Soda, Spinach Artichoke Dip, Chicken Alfredo).
- Ingredients: List of main ingredients for each menu item.
- Price: Price of the menu item.
- Profitability: Label indicating the profitability of the item (Low, Medium, High).

It has 1000 records across 4 categories: Appetizer, Beverage, Main Course, and Desserts, with 16 unique menu items as shown in Table 1. This sample illustrates how columns relate in this semi-structured dataset. The flat structure is suitable for testing LLM-based inference of hierarchies (e.g., MenuCategory $\rightarrow$ MenuItem $\rightarrow$ Ingredient), semantic grouping, and schema evolution.

Rest.ID	Category	MenuItem	Ingredients
R003	Beverages	Soda	[confidential]
R001	Appetizers	Spinach Artichoke Dip	[Tomatoes, Basil, Garlic, Olive Oil]

Table 1: Sample of Restaurant Menu Dataset

4. LLM-driven Ontology Construction Framework

Traditional ontology engineering treats knowledge modeling as a static, expert-driven process, while even LLM-based approaches often generate schemas in a one-time manner. Although these methods accelerate initial ontology creation, they fail to address the dynamic nature of evolving business data and semantic requirements [15]. In order to overcome the issues of the traditional ontology alignment systems, we proposed an agentic system that can get incorporated within BI systems and models the ontology schema and semantics of it in a continuous approach. In this framework, LLMs will be used to

[1]https://www.kaggle.com/datasets/rabieelkharoua/predict-restaurant-menu-items-profitability

create the ontology concepts and semantics and human comes into play for validation and checks. As the businesses data is changing all the time, our agentic system is continuously interacting with the data to redefine the semantics and hierarchies and suggest the changes to human for review. In Figure 1, the structure and layers of our agentic system for ontology construction is propose.

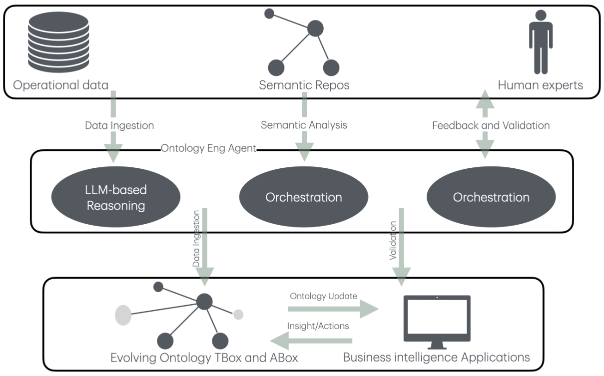

Figure 1: LLM-driven ontology construction system

The agentic ontology construction system is introducing the four main cores and capabilities: hierarchical schema design, semantic relationships identification, schema gap detection as data evolves and changes, and human expert validation. We belive that incorporating all these steps can introduce a new era for ontology engineering and transform them from static schema modeling into more dynamic knowledge management process.

Hierarchical Schema Induction

Ontology agent is designed to extract the classes, properties, and semantics from semi-structured data which is in different formats live csv but they are contains hierarchical structure, relationships and semantics. In this stage, the agent is extracting the classes and relationships between them to identify the structures and hierarchies such as **Restaurant** → **MenuCategory** → **MenuItem** → **Ingredient**. Then the agent is responsible to add them to an ontology-compatible format to formalize them as ontology. At this stage classes, objects, and their relationships are identified and formalized for ontology development. In this method, entities and structures are coming from the data directly and make it easy for the business environments and their dynamic nature which is evolving continuously.

Semantic Relationship Reasoning

Main component of each ontology is the semantic entities and relationship amongs them. The ontology construction agent needs to performs semantic understanding and reasoning to identify and extract all these hidden semantics which was not easily identified by patterns. This agent uses contextual undestanding coming from generative models to infers additional semantic objects from the data and do iterative enrichment to get into finer levels of the semantic. Example of it from our data is ingredient and description. For instance, this agent can identify the logics and relationships as

```
Parmesan → typeOf → Cheese
Cheesecake → belongsTo → Dessert
Chicken Alfredo → partOfCuisine → ItalianCuisine
```

By adding more semantics level to the ontology employing the agants, more context and logics gets added to the context and more complex analytical queries can be handled in BI systems. For example, this level of semantics can enable analysts to query a dish by their ingredients or cuisine type without requiring them to know the schema.

Schema Gap Detection and Evolution Planning

In today's BI world with continuous data generation and evolvement, it is very challenging for domain experts to maintain the schema and ontologies up to date and clean. In business and operation environments, new categories, objects, logics, and entities are frequently get added to the data as corporations or businesses introduce new products or they want to remove them. For example, if a restaurant what to add a new categories of "food", or add a part related to "food allergy". If the ontology is created manually, then updating the schema and adding the changes is going to be time consuming and expert dependent.

We tried to address this issue in our argentic framework by designing a mechanism to constantly incorporate the any changes and update the schema accordingly. In fact, this agent is monitoring the data ingestion and as soon as it observe any new entity that doesn't match with the current schema and ontology, it proposes a new change. Basically this agent identifies the gaps and propose the updates on the ontology to the system.

Imaging a new category of Vegan foods gets added to the restaurant data, the agent doesn't need to wait for domain expert to update the ontology; instead, the agent do some evaluation to understand the semantic behind this newly added term to find out whether add them to the existing MenuCategory as a new object or introduce a new entity branch representing the dietary classification. By analyzing the relationships between the current schema and new entity,the agent can ensure that any new changes to the schema is logically aligned with the rest of the ontology. In this mechanism, we can claim that ontology development is more a dynamic process incorporating human intervention and is allowing the semantic model to gets itself updated as the business data changes everyday.

Human-in-the-Loop Governance and Validation

LLMs are key components in this agentic ontology alignment system; however, they cannot be fully autonomous and be relied on in ontology construction tasks as they may introduces the risk of semantic inconsistencies, hallucination and adding conceptual noises. They are performing promising the conceptual reasoning and understanding the concepts, but any simple error can propagate into the whole data system and impact the analytics and decision making. In order to avoid such catastrophes, we are proposing the human-in-the-loop supervision to preserve the integrity and reliance while having the system to be more automated.

Domain expert are part of this system that can validate the proposed changes that the agents are generating from candidate schema and semantic relationships. Any validation method, such as rule-based constraints or numerical confidence cut-offs can also be incorporated in this system to support it. In fact, this approach the ontology engineering is a co-engineer that combining the automated pipeline with an expert there to oversee the correctness and consistency of the ontology with business semantics.

Agentic Ontology Lifecycle in BI Systems

The agentic workflows with all their features get integrated into the BI pipeline to make continuous ontology lifecycle. We propose our agentic ontology pipeline to be integrated into the business data pipeline instead of having it as an stand alone and static system that generates the ontology in a one-time demand. In this embedded architecture as the new data arrives, the ontology agents starts to analyze, propose, enrich and update the semantic layer and it also keeps the current model be aligned with any new business structures which is changing. This is one the the key feature of this agent as it offer a dynamic ontology ontology system which is adapting itself with the semantic of the data integration, analytics, readiness and knowledge management in different enterprise environment.

5. System Architecture

In this section, we explain the architecture of the agentic ontology pipeline as a modular system that integrates the ontology engineering into the BI pipeline. Indeed, the overall ontology agent is like a semantic layer which is repeatedly monitoring the data, constructing the new entities, updating the ontology as new data gets ingested. The powerful feature of this layer is that this layer is independent from the schema and can work with that interactively. This agentic system is containing five different components (modules) that are working together to form the ontology management agent; these five components are data ingestion, ontology induction, semantic reasoning, schema evolution, and human validation which are all connected to an ontology repository. Enterprise and business data gets ingested and the agents start analyzing it, new structures and semantics gets inferred by the agent and piped into the evolving ontology, which is connected to the BI system. Here is the link to agents being used in this architecture. [2].

- **Data Ingestion and Preprocessing Layer** This is the initial layer and is directly connected to the business data. It is in charge of collecting the datasets from different sources defined by the business. This data is in semi-structured formats (e.g., CSV, relational tables, JSON). This layer is doing all data normalization, cleaning, and some attribute extraction to prepare the data for semantic analysis and the next layer. (**Input**: Raw semi-structured data | **Process**: Schema normalization, attribute extraction, basic cleaning, entity identification | **Output**: Structured and cleaned dataset with defined attributes and entity fields (e.g., RestaurantID, MenuItem, Ingredients)
- **Ontology Induction Module** Infers ontology classes, relationships, and hierarchies from structured data using LLM-guided reasoning. For example, it derives relations like Restaurant → offers → MenuItem and maps them into RDF/OWL schemas. (**Input**: Preprocessed structured dataset | **Process**: LLM-guided extraction using prompting to identify entities, relationships, and hierarchies; mapping to ontology schema (RDF/OWL) | **Output**: Initial ontology with classes, object properties, and hierarchical relationships)
- **Semantic Reasoning Module** Enriches the ontology by inferring higher-level relationships (e.g., Parmesan → subclassOf → Cheese, Chicken Alfredo → cuisineType → ItalianCuisine) using contextual reasoning. (**Input**: Initial ontology + enriched textual/contextual attributes | **Process**: LLM-based reasoning and embedding similarity to infer implicit relationships, subclass hierarchies, and semantic groupings | **Output**: Enriched ontology with additional semantic relations (e.g., subclassOf, cuisineType))
- **Schema Evolution Planner** This agent is responsible for detecting data changes and ingesting the updates. In this layer, a constant data monitoring is happening according to the cadence that we define and if the data gets any changes, the schema changes will be proposed accordingly (e.g., new classes or hierarchies). Referring back to the "plant-based" foods appears in the data, this layer monitors and evaluates what would be the best approach between extending the current entities without any changes to the ontology or creating a new ontology entity.
(**Input**: Updated/new data + existing ontology | **Process**: Change detection, schema gap analysis, LLM-assisted proposal of new classes/relations or hierarchy updates| **Output**: Updated ontology schema with proposed changes and evolution)
- **Governance and Validation Interface** In this layer, domain experts can review and validate the generated or updated ontology by the agent. This is where the human-in-the-loop component gets enabled for validation and correctness analysis. Domain experts can approve or reject the changes or interact with the system for further instruction.
(**Input**: Proposed ontology updates and relationships | **Process**: Human-in-the-loop validation, rule-based checks, confidence filtering on generated entities |**Output**: Validated and approved ontology updates and making sure on semantic correctness)

[2]Prompt template in git: https://github.com/amiinio/OntologyMegaAgent/tree/main/prompt

- **Ontology Repository and BI Integration** The final knowledge graph is stored in a central repository to maintain the classes, repositories and semantic relationships. This KG repository is accessible by querying this layer to perform the analysis like (e.g., items by cuisine, ingredients by type, or dishes by dietary class) and making the whole system more flexible and enriching for knowledge management in any business intelligence and analytic system. (**Input**: Final validated ontology (RDF/OWL graph) | **Process**: Storage in knowledge graph repository; integration with BI tools for querying | **Output**: Queryable semantic layer enabling analytics and decision support)
- **End-to-End Ontology Pipeline** The overall ontology construction process which operates as a continuous pipeline:

$$
\begin{aligned}
\text{Operational Data}_{-\text{input}} &\rightarrow \text{Data Ingestion and Preprocessing}_{-\text{prompt}} \\
&\rightarrow \text{Ontology Induction}_{-\text{prompt}} \\
&\rightarrow \text{Semantic Reasoning}_{-\text{prompt}} \\
&\rightarrow \text{Schema Evolution Planning}_{-\text{prompt}} \\
&\rightarrow \text{Human Governance and Validation} \\
&\rightarrow \text{Ontology Repository} \rightarrow \text{Business Intelligence Applications}
\end{aligned}
$$

The ontology constructs an agent and its integration in data systems, the ontology construction becomes a continuous and adaptive layer within the data system rather than a one-time modeling effort.
LLM Configuration: The framework uses a GPT-4–class LLM with defined and structured prompt templates for ontology induction and reasoning, using deterministic settings (temperature = 0). Full prompt details and configurations are provided in the GitHub repository 2.

6. Evaluation

We conducted an evaluation to show the power and feasibility of our agent and evaluate the performance of it on ontology generation quality, schema updates and query support of it. We randomly selected 100 samples of restaurant records and piped that data into the proposed system. Our system inferred the main classes such as "Restaurant", "MenuItem", "Ingredient", and "MenuCategory". It also inferred the hierarchical model with 5 classes, 5 object properties and 360 triples (A-box) as full schema coverage.

Out of 360 triples, 356 were correct and 4 were incorrect, yielding Precision = 0.997, Recall = 1.0, and Accuracy = 99.7%. AI models will always get involved with the hallucination issue causing by data quality or representation. For example, the triple in our data as "the triple Grilled_Steak → containsIngredient → Fettuccine" is semantically an incorrect inference and entity which is originating from the mislabeled data point in our original dataset. This observation is showing that although the semantic extractions can be reliable and reasonable, the system needs the human intervention and consistent semantic checks although incorporating schema constraints and automated validation mechanism are necessary.[3]

- **Ontology Generation** We tested the ontology generation capability on out 100 samples to check the agent performance on generating ontology from semi-structured CSV data. The agent generated the key classes like "Restaurant", "MenuItem", "MenuCategory", "Ingredient" and it generates the hierarchical structure and model with 5 classes, 5 object properties, and 360 triples which is achieving full schema coverage. We also ran a manual validation of 360 record of triples to investigate the recall and overall coverage. Manual validation shows 356/360 triples are correct, with 4 errors caused by data inconsistencies, for example, wrong ingredient record in the original data source. Although the result showed very promisssing performance on the main ontology

[3]Evaluation data: https://github.com/amiinio/OntologyMegaAgent/tree/main/evaluation

construction agent, the mistakes and noises made us confirm the need for human validation layer in this pipeline.

- **Schema Evolution Capability** One of the main features in our system is its ability to evolve and adapt with new data changes. We also wanted to simulate the introduction of a new data record, like category and attribute into the system to be able to evaluate the feature. Our observation also confirmed that by adding a new record of data including a new category, the ontology evolution planner identified the schema gap and proposed a new update to the ontology. For example, by introducing a new record including the "Plant-based" food, the system proposed a new subClass semantic relationship withing the ontology. This result, showed that this agentic system is capable of adapting to any data changes and accordingly offering new ontology changes.
- **Semantic Query Support** The integration of the otologies as knowledge graph into a repository will enable the BI users or systems to query the semantics and relationships directly which is more stronger than the raw table querying. For example, the user can query the data with cuisine, or dish name, or ingredients in that dish. Having semantic ontologies as background knowledge is unlocking more insight from the semi-structured data.

7. Conclusion and Future Work

In this paper we presented an AI-based Agentic framework with automate the process o ontology development for enterprise and business intelligent environments. This system will be integrated into the BI system for generating the ontologies, evolving them as the data changes, and proving the platform to query them directly. By combining all agentic layers of this framework as ontology induction, semantic extraction and schema evolvement with the human-in-the-loop agent, this system can transform a semi-structured data into a dynamic knowledge graph. Our prototype on restaurant data showed the system ability to support all the layers and generated a meaningful ontology validated by human. This effort helped reducing the manual work on ontology modeling.

Our ongoing research is still related to the use of generating model for agentic ontology evolvement in both T-Box and A-Box level. We are testing our model in different use cases and larger datasets. We are also adding guardrails into the evaluation agent to make the human involvement more focused.

Declaration on Generative AI

The authors used generative AI tools (ChatGPT 4.o) during the preparation of this manuscript, primarily for language refinement and grammar and spelling checks. All technical content, analysis, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of this paper.

References

[1] A. Mikroyannidis, B. Theodoulidis, Ontology management and evolution for business intelligence, International journal of information management 30 (2010) 559–566.

[2] R. Amini, S. S. Norouzi, P. Hitzler, R. Amini, Towards complex ontology alignment using large language models, in: International Knowledge Graph and Semantic Web Conference, Springer, 2024, pp. 17–31.

[3] M. C. Suárez-Figueroa, R. García-Castro, B. Villazón-Terrazas, A. Gómez-Pérez, Essentials in ontology engineering: Methodologies, languages, and tools (2011).

[4] H. Saggion, A. Funk, D. Maynard, K. Bontcheva, Ontology-based information extraction for business intelligence, in: International Semantic Web Conference, Springer, 2007, pp. 843–856.

[5] Z. Bi, J. Chen, Y. Jiang, F. Xiong, W. Guo, H. Chen, N. Zhang, Codekgc: Code language model for generative knowledge graph construction, ACM Transactions on Asian and Low-Resource Language Information Processing 23 (2024) 1–16.

[6] N. F. Noy, D. L. McGuinness, et al., Ontology development 101: A guide to creating your first ontology, 2001.
[7] A. Maedche, S. Staab, Ontology learning for the semantic web, IEEE Intelligent systems 16 (2005) 72–79.
[8] P. Cimiano, Ontology learning and population from text: algorithms, evaluation and applications, Springer, 2006.
[9] G. K. Quirino, M. P. Barcellos, R. A. Falbo, Opl-ml: A modeling language for representing ontology pattern languages, in: International Conference on Conceptual Modeling, Springer, 2017, pp. 187–201.
[10] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, arXiv preprint arXiv:2403.08345 (2024).
[11] A. Oyewale, T. Soru, Llm-driven ontology construction for enterprise knowledge graphs, arXiv preprint arXiv:2602.01276 (2026).
[12] S. Sadruddin, J. D'Souza, E. Poupaki, A. Watkins, B. Karasulu, S. Auer, A. Mackus, E. Kessels, Schema-miner pro: Agentic ai for ontology grounding over llm-discovered scientific schemas in a human-in-the-loop workflow, Semantic Web Journal (2025).
[13] X. Wang, B. Li, Y. Song, F. F. Xu, X. Tang, M. Zhuge, J. Pan, Y. Song, B. Li, J. Singh, et al., Openhands: An open platform for ai software developers as generalist agents, arXiv preprint arXiv:2407.16741 (2024).
[14] R. Sapkota, K. I. Roumeliotis, M. Karkee, Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges, Information Fusion (2025) 103599.
[15] S. L. Borowicc, S. N. Alves-Souza, Semi-automatic domain ontology construction: Llms, modularization, and cognitive representation, Proceedings Copyright 64 (2025) 73.