Ontology-driven context engineering for semantically-aware chatbot responses

Semantic Web Journal
XX(X):1–20

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Mateus Peixoto[1], Gabriel Silva[1], Lucas Maddalena[1], and Fernanda Baiao[1]

Abstract

Large Language Models (LLMs) have achieved notable success across diverse domains, yet they remain prone to hallucinations and omissions due to their probabilistic reasoning and black-box nature. These limitations are particularly critical in high-stakes domains, where inaccurate or incomplete responses may lead to severe consequences, for example, failing to instruct a client correctly in an account activation procedure, despite having access to the information. Recent efforts to mitigate such issues have focused on integrating structured knowledge into LLM pipelines, predominantly through knowledge graphs. However, the selection of the relevant entities to generate a response may not be intuitive to retrieve from the graph's structure alone. The role of deeper semantic structures has not been systematically explored. Specifically, ontologies with axiomatic foundations could be used for automated inference, leading to a more precise set of entities and, therefore, potentially providing a better input to generate more accurate LLM responses. This paper investigates the impact of deep semantic enrichment on LLM-based knowledge extraction and response generation. We situate our contribution within the Neuro-Symbolic AI paradigm and propose a framework that combines knowledge graphs for contextual storage with ontological reasoning grounded in structural patterns that are ontologically well-founded in UFO. By leveraging the well-founded semantic nature of ontological entities, rather than solely their graph-like structure, the proposed approach constrains interpretation and supports more precise reasoning during prompt construction. The framework is evaluated by a real company using a domain ontology of the Brazilian financial system, enriched with business rules and processes provided by an industry partner. We compare scenarios with varying strategies for leveraging semantic knowledge from the ontology, ranging from light-semantics approaches that rely solely on graph-based properties to a problem-based baseline designed by domain experts. Experimental results demonstrate that the Deep Semantics approach achieves superior performance in both entity extraction and response generation quality, consistently extracting a higher number of relevant entities and producing more complete and semantically aligned responses. These findings highlight the benefits of incorporating ontological depth and axiomatic semantics into LLM workflows. Overall, this work provides empirical evidence that deeper knowledge representations can significantly enhance LLM reliability and interpreting ability, advancing the development of knowledge-driven large language models.

Keywords

Ontology, Ontological Unpacking, Chatbot, Knowledge Graph, Deep Semantic Enrichment, Context Engineering

Introduction

LLMs have demonstrated a remarkable ability to understand and generate natural language texts, therefore often used as chatbots to interact and provide correct responses to users and clients. However, they remain prone to hallucinations —producing factually incorrect information. While ensuring high-quality training and reference data is vital to mitigating these errors, domain-specific data selection is equally important, as ambiguity can severely impact accuracy. This paper extends prompt engineering techniques for knowledge extraction by incorporating well-founded ontologies, enabling more effective reasoning through deeper semantic structures. By combining knowledge graphs for textual context storage with ontological unpacking for entity-level inference, our approach supports precise knowledge extraction and response formulation.

This work is situated within the broader context of Neuro-Symbolic (NeSy) AI, which integrates deep learning with symbolic reasoning. NeSy seeks to overcome the individual limitations of symbolic and sub-symbolic paradigms while leveraging their strengths—uniting two fundamental aspects of intelligent cognition: the ability to learn from experience and the capacity to reason based on learned knowledge.

Large language models (LLMs) have been successfully employed across various applications, such as natural language understanding and generation, and across a myriad of domains, such as nuclear medicine (Alberts et al. 2023), the legal domain (Savelka 2023), and tourism (Carvalho and Ivanov 2024). However, these models face criticisms due to their reliance on memorizing information from training data, rather than accurately understanding and recalling facts

[1]Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil

Corresponding author:
Fernanda Baião, Department of Industrial Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil.
Email: fbaiao@puc-rio.br

put into context. This often leads to hallucinations, where the models generate outputs that contain factually incorrect or out-of-context statements (Sallam 2023; Deng and Lin 2023).

Another problem associated with LLM-generated answers is the lack of accountability, as presented by Zuccon et al. (2023). LLMs provided 86% of non-existing references, and of the remaining 14% of references that truly exist, a large part of them do not support the claims of the response. Therefore, there are various issues regarding the current performance of LLMs that stem from a lack of knowledge absorption capability, generating spurious correlations in an attempt to provide responses that emulate previously approved answers, regardless of the semantics of the generated content.

The black-box nature of LLMs also poses challenges in terms of human interpretation and explainability, as they implicitly represent knowledge within their parameters, making it difficult for human users to understand, assess, or audit the acquired knowledge. Furthermore, LLMs perform reasoning through probabilistic processes, which may yield uncertain or indecisive outcomes. Even though some LLMs employ techniques like chain-of-thought to explain their predictions Wei et al. (2022), these reasoning methods may still suffer from hallucinations, biases, and interactions with maliciously altered data, leading to unforeseen distortions in the model's response.

The limitations of LLMs become particularly concerning in high-stakes scenarios, such as medical diagnosis, or as they are widely employed as question-answering systems for customer assistance in legal and financial domains. In these contexts, an incorrect diagnosis or prediction may pose severe consequences. Explanations provided by LLMs may contradict established medical or legal knowledge. Hallucinations are of great importance in this regard and are typically discussed in the literature. However, a subtler problem may arise in the form of omissions, as was the case presented by Birkun and Gautam (2023), where the LLM did not show vital information, not capturing the complexities of the concepts involved.

McKenna et al. (2023) conducted a detailed analysis of the potential causes for hallucinations, by which they concluded that corpus-based biases are a major source for hallucinations, impacting word frequency. Corpus-based bias could be better described as the overlap between requests and the provided training data, resulting in a response with no relevant information or contradictory information, depending on the direction of the query. Therefore, the information selected to provide context is considerably relevant for LLM performance as chatbots.

Both knowledge graphs and ontologies serve as symbolic representations of knowledge (Zhang et al. 2021). Since symbolic reasoning often struggles with ambiguity and context, combining symbolic and machine learning-based approaches is a promising path forward. Symbolic AI, while precise, tends to be less efficient computationally (Shakarian et al. 2023) and less flexible with out-of-context inputs. LLMs, conversely, excel at generalization (Yang et al. 2024), but lack semantic grounding. The axiomatic nature of ontological entities can constrain user interpretation (Vasilecas 2007), and we argue that integrating this knowledge into LLMs could help limit inappropriate or inaccurate outputs.

Significant progress has been achieved in integrating structured knowledge into text generation pipelines as a potential solution to reduce hallucinations. However, most studies rely exclusively on knowledge graphs. As shown in the comparative study by Cadeddu et al. (2024), while injection techniques differ in strategy and performance, the underlying knowledge structure often remains unchanged. This reveals a gap in the literature: the impact of the depth of the knowledge structure—particularly the added axiomatic richness of ontologies—has yet to be systematically explored.

Knowledge structures have made essential contributions to the field of Natural Language Processing (NLP). Recently, integrating knowledge graphs with large language models (LLMs) has been shown to enhance performance across various tasks. However, growing attention has been directed toward more sophisticated forms of knowledge representation, such as ontologies.

Ontologies offer a semantically rich representation of expert domain knowledge, incorporating structural elements—like classes and relations—common to knowledge graphs, while adding formal axioms that define domain constraints (Guarino et al. 2009).

Despite recent progress, many approaches still overlook the value of deeper knowledge structures, and confusion between the terms **knowledge graph** and **ontology** remains prevalent in the literature—an issue that can hinder the field's development.

This work explores the state of the art, identifies emerging patterns, and clarifies distinctions between different forms of structured knowledge and their benefits for various LLM tasks. By examining these aspects, we also pinpoint the challenges and underexplored areas in advancing toward knowledge-driven large language models.

The remainder of this paper is organized in the following sections. The Literature review section explains the main differences between the conceptualization of knowledge graphs and ontologies and describes key foundational ontology patterns applied to our deep semantics proposal. The Ontology-driven semantic enrichment section describes the proposed framework in detail. The Experiment Design section shows the results of assessing the proposed framework using the various algorithms devised, as a way to grasp the effectiveness of the deep semantics proposal and the data analyzed. The Results section details the performance of each algorithm in knowledge extraction and response generation quality. Finally, the Conclusion summarizes the main findings and contributions, presents some closing remarks, and discusses future work.

Literature review

This section is envisioned to standardize the conceptualization of ontologies and knowledge graphs, explain the differences between ontologies and well-founded ontologies, and detail some of the main Foundational Ontology patterns, whose proper comprehension is paramount to establishing the proposal of Deep Semantic Enrichment. Properly comprehending the differences among the knowledge structure

approaches adopted is key to understanding how semantically deeper knowledge structures, i.e., constructed based on a well-founded common universal conceptualization, can benefit more precise inference for various tasks, such as automated context injection for response-generating prompts.

Knowledge Graphs

A Knowledge Graph (KG) is a data structure that has become integral to various applications in machine learning, natural language processing, and semantic web technologies. At its core, a KG represents information in a structured format, typically using RDF, the Resource Description Framework, as defined by Ferguson and Hebels (2003), where knowledge is expressed as a set of triples; each triple consists of a head (subject), relation (predicate), and tail (object), such as (Albert Einstein, Winner Of, Nobel Prize). This structure allows for the representation of factual knowledge in a directed graph format, where nodes symbolize the entities denoting the subject and the object of the triple, and edges depict the relations between these entities. The research community often uses the terms "knowledge graph" and "knowledge base" interchangeably, depending on the context and specific characteristics emphasized in research discourse. Several scholars have attempted to outline what constitutes a KG. For example, Ehrlinger and Wöß (2016) focuses on the reasoning capabilities that KGs facilitate, highlighting their utility in AI applications that require inference and logical deduction. While Wang et al. (2017) contributes to the definition by describing a KG as a multi-relational graph that consists of entities and relations, treated as nodes and edges, respectively.

Wang et al. (2017) defines a KG as a multi-relational graph $G = \{E, R\}$, where E represents entities (nodes), R stands for relations (different types of edges), and each edge represents a fact as a triple of the form (head entity, relation, tail entity) indicating that the head entity is connected to the tail entity by a specific relation, e.g., (AlfredHitchcock, DirectorOf, Psycho). The relations within a KG enhance its utility for complex querying and reasoning tasks. One inherent characteristic of KGs that poses both challenges and opportunities for further research is their incompleteness. No knowledge base can claim to encompass all knowledge comprehensively; new entities, facts, and relationships continue to emerge, necessitating ongoing updates and revisions. Therefore, the systems must be constructed following the "Open World Assumption"(OWA). Under OWA, the knowledge base is considered inherently incomplete, meaning that the absence of information does not imply its negation (Lutz et al. 2012). Therefore, systems utilizing KGs require robust mechanisms for learning, updating, and possibly inferring missing information to maintain their relevance and accuracy over time. Furthermore, the dynamic nature of knowledge creation means that KGs must be adaptable and scalable to accommodate new information efficiently and effectively.

In conclusion, while the precise definition of a knowledge graph may vary slightly among different scholars, the fundamental concept remains consistent: a graph-based representation of real-world entities and their interrelations, structured in such a way as to facilitate efficient processing, querying, and reasoning.

Ontologies

Ontologies have been used in several fields within Computer Science and correlated areas, especially due to their increasing adoption in interdisciplinary initiatives and exploring foundational abstraction levels involving cognitive psychology, philosophy of language, and linguistics (Guizzardi 2005). Therefore, it is fair to affirm that there is no consensual definition embracing all its usage contexts.

A commonly used definition in the literature and the basis for this systematic review is the one in Gruber (1993): An ontology is a formal, explicit specification of a shared conceptualization. In this work, conceptualization refers to an abstract model of the world or a specific domain, explicit implies that the elements of the ontology must be clearly defined, and formal indicates that the ontology should be machine-processable. Despite the definition not being recent, it encapsulates the essential aspects that differentiate an ontology from other knowledge structures, as will be further presented in this section.

Gruber advances the idea that an ontology is essentially the representation of the knowledge of a domain, involving a set of objects and their relations, described by a specific vocabulary. The formalization of the ontology, according to Gruber, consists of a logical theory based on an ontological commitment and a vocabulary that seeks to approximate the intended models of a domain as specified by that commitment. On the other hand, Maedche and Staab (2001) offers a complementary definition that most existing ontology representation languages are consistent with, such as Resource Description Framework (RDF) and Ontology Web Language (OWL). Maedche and Staab (2001) proposes that an ontology should be described by a five-tuple O = $\{C, R, CH, rel, OA\}$, where:

- **C** and **R** are two disjoint sets, called the set of concepts and the set of relations, respectively.
- $CH \subseteq C \times C$ is a concept hierarchy or taxonomy, where $CH(C_1, C_2)$ indicates that C_1 is a subconcept of C_2.
- $rel : R \rightarrow C \times C$ is a function that relates the concepts non-taxonomically.
- **OA** is a set of ontology axioms, expressed in an appropriate logical language.

In short, ontologies represent a fundamental tool for knowledge representation and structuring. They are structured as sets of concepts, relations, instances, and axioms, enriched by an ontological commitment that ensures the precise and rich definitions of a domain. This structure not only represents the concepts and relations of a domain but also supports logical and deductive inference, allowing adaptability and the extraction of new knowledge in a structured manner. Thus, it is imperative to investigate the benefits that ontologies and other forms of structured knowledge may provide to Large Language Models, which, a priori, are not capable of detecting the entities of the real world that texts refer to and are prone to hallucinating. Another rich aspect of ontologies to consider is the hierarchy of generalization, which categorizes ontologies into different levels based on scope and abstraction levels. This hierarchy consists of the upper level, task, domain, and application ontologies, each

serving a different level of generalization, as depicted in Figure1.

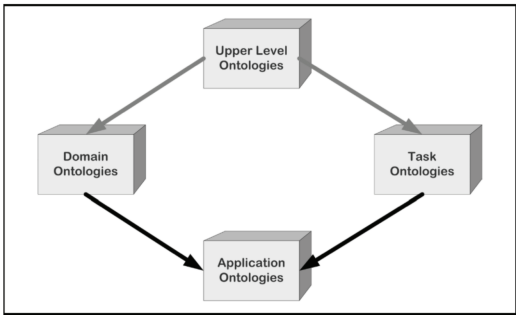


Figure 1. Ontology classification by Guarino (1998)

Guarino et al. (2009) proposes the following definitions for each level:

- *Upper-level ontologies* represent very general concepts - such as space, time, matter, event -, which are independent of a particular domain or problem.
- *Domain ontologies* and *task ontologies* represent the vocabulary related to a generic domain or generic task or activity, by deepening and specializing the terms introduced in the upper-level ontology
- *Application ontologies* describe concepts that depend both on a domain and a task. He points out that the concepts often correspond to roles played by domain entities while a certain activity occurs.

By acknowledging these levels of generalization, ontologists can better design ontologies that are both flexible and contextually appropriate, promoting more effective data interoperability, knowledge sharing, and decision-making support across various systems and applications. This multi-tiered approach is fundamental in addressing the complexity of real-world applications, where the specificity of information varies greatly across different contexts.

The language utilized for the construction of the ontology is paramount. In this paper, we adopted OntoUML as an ontologically well-founded language for ontology-driven conceptual modeling, designed as an extension of UML (Unified Modeling Language). It incorporates ontological distinctions from a foundational ontology to ensure that conceptual models reflect real-world semantics accurately. OntoUML was developed based on the Unified Foundational Ontology (UFO), which is a foundational ontology created by Guizzardi (2005) that integrates theories from formal ontology, cognitive science, linguistics, and philosophical logic into a coherent ontological framework. By utilizing OntoUML and UFO, the system is capable of making more complex inferences, having potential gains for entity selection for LLM response generation.

RAG approaches relying on structured knowledge representations

It is worth noting that recent work has increasingly recognized the limitations of text-centric Retrieval-Augmented Generation (RAG) for knowledge-intensive tasks requiring structured reasoning. Standard RAG approaches retrieve flat, semantically similar text chunks, which limits their effectiveness when relational or ontological constraints must be respected (Edge et al. 2024; Sharma et al. 2025). To address this, several lines of work have proposed grounding retrieval in structured knowledge representations (herein referring either to knowledge graphs or to ontologies). Edge et al. (2024) proposes GraphRAG, which uses a large language model to derive an entity knowledge graph from source documents and then generates community summaries over entity clusters, leading to substantial improvements over conventional RAG baselines on global, corpus-wide queries. Sharma et al. (2025) introduce OG-RAG, which constructs a hypergraph representation of domain documents where each hyperedge encapsulates clusters of factual knowledge grounded in domain-specific ontologies; their evaluations show a 55% increase in accurate-fact recall over standard retrieval. More broadly, the integration of knowledge graphs into RAG pipelines has been surveyed and benchmarked by Chen (2025), who highlight that structured retrieval enables more robust multi-hop reasoning and reduces hallucinations in settings where flat text retrieval fails.

Fundamental differences between ontologies and knowledge graphs

The distinction between ontologies and knowledge graphs is essential in the domain of knowledge management and information science. While both structures aim to organize and represent knowledge, they do so with fundamentally different approaches and purposes. This subsection elucidates these differences by exploring the intrinsic characteristics of each, emphasizing their unique contributions to the field.

Ontologies are inherently philosophical, grounded in a desire to establish a universal and meticulously formal vocabulary for a particular domain. Ontologies facilitate a common understanding and enforce a rigorous structure that enhances systematic reasoning and inference capabilities. The architecture of ontologies often incorporates a hierarchical system of classification that ranges from foundational to application-specific layers. This methodological stratification allows for the delineation between general and specialized knowledge, enabling precise modeling of domains. Such a structured approach not only aids in human understanding but also enhances machine processability, thereby supporting complex logical operations and the deduction of new knowledge from facts (Maedche and Staab 2001).

Conversely, knowledge graphs prioritize practical usability and computational scalability over philosophical depth, focusing on the management and utilization of large-scale data. KGs are graph-based structures where nodes represent entities and edges depict the relations between these entities. This format is exemplified in the Resource Description Framework (RDF), which organizes knowledge in triples (head, relation, tail) that collectively form a navigable graph (Paulheim 2017).

Unlike ontologies, KGs are less concerned with strict hierarchies and are more focused on the breadth and interconnections of data. They excel in environments where rapid querying and efficient data retrieval are crucial. KGs handle a vast array of relations and entities, offering a flexible

and dynamic framework that can adapt to the continuous influx of new information. This adaptability makes KGs particularly effective for applications in artificial intelligence and machine learning, where they can quickly integrate and analyze large datasets. The core distinction between ontologies and KGs lies in their targeted utility. Ontologies provide a rich, in-depth understanding of concepts, which is indispensable in scenarios requiring detailed semantic analysis and rigorous logical deductions. In contrast, KGs are better suited for handling extensive datasets where the interconnection and accessibility of information are more significant than the depth of knowledge.

Foundational Ontology Patterns in OntoUML

A Foundational Ontology pattern (FOP) in OntoUML is a structure that reflects the micro-theories put forth by its underlying foundational ontology. UFO specifies a system of micro-theories that addresses the main classic conceptual modeling concepts, including Types and Taxonomic Structures, Attributes and Weak Entities, Datatypes, Relations, Parthood, Events, and Higher-Order Types (Power Types), among others.

The ontological distinctions formally specified in UFO are represented as modeling constructs in OntoUML (UFO´s isomorphic representational language) with their corresponding axiomatization. They serve as a guideline for building OntoUML models that are syntactically compliant with UFO constraints. In other words, FOPS are the modeling primitives in OntoUML, reflecting UFO´s underlying ontological micro-theories. As a consequence, OntoUML is a pattern language whose models are constructed via the combined instantiation of FOPs (Ruy et al. 2017). A FOP is a domain-independent structure derived directly from UFO, while a Domain-Related Ontology Pattern (DROP) is its domain-specific counterpart: a recurrent configuration of classes and relations that reuses and specializes the foundational patterns within a particular domain (Falbo et al. 2013)

Therefore, understanding a conceptual model by means of its composing FOPs and DROPs is crucial for an in-depth comprehension of the proposed framework, as they provide the rationale underlying the applied semantic enrichment algorithm. Such structural design patterns (FOPs and DROPs) are made explicit by OntoUML stereotypes, which guide the algorithm in identifying which entities (and relationships) contribute to understanding a given entity. This understanding helps determine which information is useful for answering a specific question. Hence, we argue that a chatbot may benefit from structural design patterns in assessing the relevance of each entity based on its ontological nature, potentially increasing the likelihood of responses referring only to entities that are relevant to answer the user's question.

The MODE pattern characterizes a MODE Universal that is linked to an ENDURANT Universal Expression through an existential dependence (inherence) relation. This ENDURANT Universal Expression serves to specify the universals whose instances act as bearers of the instances of the MODE Universal. Because a MODE may be externally dependent, the MODE pattern also includes a potentially empty set of external dependence relations that associate instances of the MODE Universal with the entities that constitute their sources of external dependence, such as linking a commitment to the entities to which that commitment refers (Ruy et al. 2017). Therefore, whenever a mode is a relevant entity, the object that it inheres in must be extracted for the proper context to be constructed.

The RELATOR pattern is defined in two variants. In the first variant, RELATORS are represented as being connected, through mediation relations—that is, existential dependence relations—to one or more textscsubstantial Universals, whose instances correspond to the entities mediated by the given RELATOR. In the second, more comprehensive variant of this pattern, a MATERIAL relation is also specified as being derived from the corresponding RELATOR Universal and established among the related mediated by instances of that RELATOR. For example, the relation married-with can be derived from the RELATOR Universal Marriage and used to connect the ENDURANT Universals Husband and Wife, whose instances are themselves mediated by instances of Marriage. As captured by the RELATOR Pattern, the resulting MATERIAL relation may have any arity, provided it is greater than two (Ruy et al. 2017). Therefore, if a given relator is relevant to comprehend the context of a situation, all the entities mediated by it are also relevant.

The COLLECTIVE pattern characterizes a COLLECTIVE Universal together with the Universals whose instances constitute the members of these collectives. Owing to the so-called WEAK SUPPLEMENTATION PRINCIPLE, which applies to all parthood relations, the sum of the minimum cardinality constraints on the member side must be equal to or greater than two (Ruy et al. 2017). Therefore, two inferences can be made: an entity whose stereotype is a collective is formed by a set of members, meaning that the members are relevant as they carry the identity principle of the COLLECTIVE. Since it is known that a KIND constitutes the sole terminal symbol in the Onto-UML grammar. That is, any recursive chain of definitions for these expressions in an OntoUML model terminates only upon reaching a KIND construct; therefore, to properly comprehend all the aspects of a collective, all the information that is inherited by its members up to their ultimate KIND must be collected; otherwise, the collective could not have its identity principle established.

The establishment of an entity KIND as the terminal symbol, some other patterns are displayed by SUBKINDS, PHASES, and ROLES stereotypes. The SUBKIND Pattern occurs in two variants. In the first, a single SUBKIND specializes a RIGID SORTAL Expression. In the second, a SUBKIND generalization set groups a disjoint (and optionally complete) collection of SUBKINDS that all specialize the same Universal, again defined by a RIGID SORTAL Expression. A SUBKIND is restricted to specializing only a RIGID SORTAL. The recursive nature of these patterns ensures that every SUBKIND, either directly or indirectly, specializes a SUBSTANCE SORTAL that supplies a uniform principle of identity for its instances (Ruy et al. 2017). Therefore, the ultimate KIND that gives the identity principle of the entity must be collected.

An ANTI-RIGID SORTAL Expression corresponds either to an instance of the PHASE Pattern or to an instance of the ROLE Pattern. The PHASE Pattern is defined by a phase

partition, that is, a disjoint and complete set of two or more complementary PHASES specializing the same SORTAL, as specified by a SORTAL Expression. Once again, the recursive definition of this pattern guarantees that a SUBSTANCE SORTAL providing a common principle of identity for the instances of these PHASES is explicitly represented in the model (Ruy et al. 2017). Therefore, whenever a PHASE exists, at least one more entity with the same stereotype exists linked to the same ultimate KIND, thus all phases that fit these criteria must be collected as well as the ultimate KIND that grants the PHASES' identity principle.

Similarly, in the ROLE pattern, a ROLE specializes a SORTAL Universal, also defined through a SORTAL Expression. However, since ROLES are relationally dependent Universals, each ROLE must additionally participate in an occurrence of the RELATIONAL DEPENDENCE Pattern. The RELATIONAL DEPENDENCE Pattern is a composite pattern that characterizes the relational dependence condition of a relationally dependent Universal. Thus, a ROLE is specified by a RELATIONALLY DEPENDENT. This relational dependence is typically captured by a RELATOR pattern, where the relationally dependent Universal appears as one of the mediated types (Ruy et al. 2017). This means that whenever an entity whose stereotype is a ROLE is relevant, the RELATOR associated with it must be extracted, along with any other ROLES mediated by the same RELATORS, and for every ROLE, its ultimate KIND must be extracted due to the identity principle inherited by it.

The CATEGORY pattern can emerge in two alternative ways. In the first, a CATEGORY directly serves as a common abstraction over two or more disjoint RIGID SORTAL expressions. In the second, a CATEGORY functions indirectly as a common abstraction over another MIXIN Expression—either through a recursive occurrence of the CATEGORY pattern itself or via an instance of the ROLEMIXIN pattern—which is ultimately associated with a set of SORTAL Expressions (Ruy et al. 2017). Therefore, CATEGORIES are common abstractions of some aspect shared by disjoint objects, such as a chair and a large wooden box, being objects that a human can use to sit on; the other objects belonging to this category may be of interest for context extraction. The ROLEMIXIN is similar to CATEGORY, the difference being that it abstracts a common aspect of two entities that are not disjoint. An example would be the ROLES of a corporate client (business) and a personal client (a person), which can be abstracted into the ROLEMIXIN of a client; this implies that any properties that apply to a client also apply to a corporate client and personal client, providing a new set of entities relevant for context.

To increase understandability, let us consider the exemplary instance of a situation that describes the marriage DROP, in which "Mary is married to John, therefore Mary is a wife". Figure 2 illustrates a simple conceptual graph that could be derived from this situation.

To represent this situation, we assume a conceptual model well-founded in UFO instantiating some FOPs and DROPS as illustrated in Figure 3.

As it is known that a wife is a role, the FOP role can be applied, and there is a relator that links it to another role, as marriage is a domain that has already been modeled. The marriage DROP already exists, stating that a person who

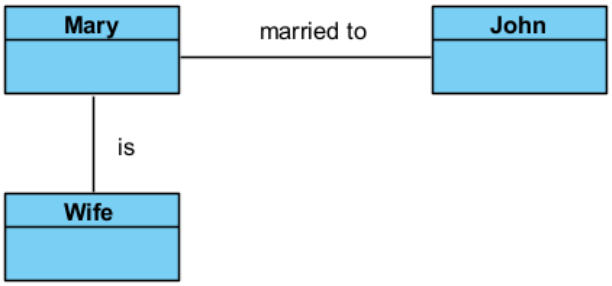


Figure 2. Marriage conceptual model

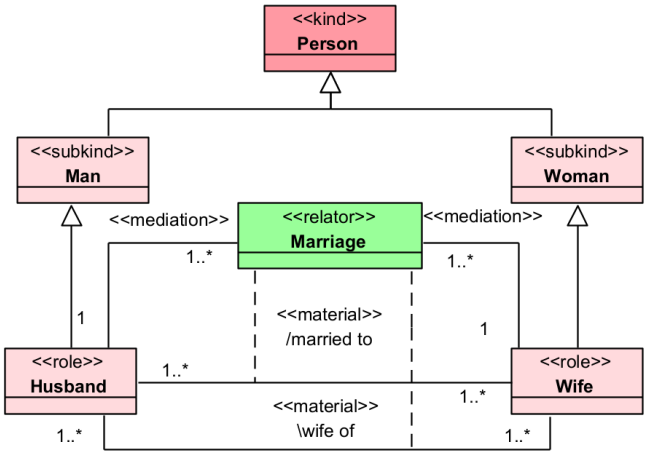


Figure 3. Marriage model applying relator and role FOPs

plays the role of wife is married to someone who plays the role of husband; therefore, Mary, being married to John, is an instance of Wife, John is then an instance of husband, specifically the one related to Mary. The DROP states that marriage is between a Husband and Wife, where Husband is the role played by a Man and a Wife, a role played by a Woman. Man and Woman are subkinds, whose ultimate kind that grants the identity principle to them is Person. It can be inferred that John is Mary's husband because the Marriage relator exists and mediates two role instances: a Husband and a Wife. If Mary, an instance of Woman, participates in a Marriage relator by instantiating the role Wife, then the same relator must mediate another participant instantiating the complementary role Husband. Therefore, if John, an instance of Man, is the participant occupying that complementary position, John necessarily instantiates the role Husband within that Marriage relator. The material relation married to between John and Mary is the basis of this inference.

These patterns summarize the main contributions that well-founded ontologies bring to context extraction, and their application is precisely the proposal for the deep semantic enrichment for context engineering for the LLM, thus providing a response-generating system that evolves from a merely Neuronal AI system to a Neuro-Symbolic AI System. Other relevant aspects are EVENTS, entities that typically alter the state of the world described by a SITUATION into another, meaning that both the SITUATION that triggers the event and the one that is brought about by it are relevant, along with all entities that participate

in the EVENT itself. Specific association stereotypes are also very relevant, such as the historical dependence and external dependence between entities, as they are essential to construct explanations.

Ontology-Driven Semantic Enrichment

Our proposal comprises a WFO-driven semantic enrichment approach, called Deep Semantics, that uses ontologies to leverage LLM-based question-answering systems, to generate correct and semantically precise responses. Any form of knowledge representation that seeks to describe reality requires some level of ontological commitment, that is, an assumption about which entities exist and are relevant for the given domain being explored, how they relate, and what structural patterns emerge from the nature of the entities and relations. By having a formal WFO pattern language, relevant information can be extracted simply from the access to semantic-driven inference, in other words, because there is a well defined structure that defines the nature and relations among the stereotypes of entities, relevant complementary information can be inferred, not just from the structure, as a knowledge graph would, but actually from the foundational design patterns of the ontology, established by semantic formalism.

Considering that correct answers are precisely what chatbots are expected to provide when interacting with the user, the models would require an ontological level of understanding to grasp what the relevant aspects are for the principle of truth-making, defined by Guizzardi and Guarino (2024) as the foundation for explainability. A truth-maker is the relation that establishes a given fact as being true, for instance: If John is married to Jane, that means each plays the role of husband and wife, respectively; however, these roles only exist after a matrimonial relator is established. Therefore, marriage establishes their roles (Guizzardi and Guarino 2024). This means, for instance, if a system must answer who John's Wife is, it must comprehend that the relator's marriage is what establishes the counterpart role of wife to Jane; if a divorce event were to occur, this relator would be terminated, leading to a different response. Thus, explanations arise from identifying the foundational truth-making structure and entities manifested in the patterns described in a given knowledge database; this is the symbolic component of the NeSy system.

This paper introduces a framework for the semantic enrichment of chatbot responses, presenting mainly the approach of Deep Semantic enrichment, which utilizes the inherent patterns of well-founded ontologies to determine the appropriate context injection for the prompt to generate an ideal response, thus creating a NeSy system of response generation. The proposed framework comprises the following components:

1. Data gathering
2. Error classification
3. Domain expert reference response creation
4. Ontology engineering and verbalization
5. Conversation partitioning
6. Entity linking
7. Semantic enrichment
8. Prompt generation
9. Response evaluation

The goal of the data-gathering component is to collect logs of user interactions with the chatbot that resulted in a failed service experience. These failed experiences relate to any situations in which the original RAG-infused documents given to the chatbot contained the necessary information to solve the problem at hand; however, the response obtained at the time was either factually wrong or responded within the assumption of a wrong context, being factually correct but irrelevant for the problem's solution. A paramount aspect of this component is the interaction with domain experts, detailing the exact points where errors were manifested in each conversation log.

The Error Classification component splits the errors by type and is paramount for the domain expert reference response creation, as it reduces the scope by eliminating errors classified as complex or associated with insufficient information derived during the data gathering component. This allows for a precise understanding of the desired topics that the reference should contain. There are five potentially overlapping error categories used for this classification, based on the exact points of error, and summarized as follows:

- Problem's nature comprehension - This was one of the most prominent types of errors, it occurred whenever, despite having enough information in the provided reference documents, the chatbot responded in another context, essentially an error derived from ambiguity, for instance, if a client informed that he is having some sort of visual glitch with the screen of his device and the chatbot responded that he should verify if the printer has any paper in it. Despite both being potential problems, they are of completely different natures.
- Hallucination - Hallucinations occur whenever the chatbot generates any response with factual errors.
- Complex errors - Complex errors are essentially the errors that originated in situations where a hypothetical correctly generated response would require a knowledge database beyond the scope of a domain-level ontology, for instance, if a customer requests an analysis regarding an accident on a given highway would impact its delivery process. This situation would require access to real-time traffic estimates, a knowledge of the company's internal delivery schedule, and initial routing. Therefore, any response classified with this type of error was deemed outside of the scope for a single domain ontology and was not fitting as input for the subsequent components of the framework.
- Operational rules - This classification of errors is from a specific misunderstanding of internal operating rules and procedures given the particular context, this is relevant as a way to differentiate fundamentally mistaken responses, such as declaring that a driver's license is a fitting document for international travel by plane, from particularly incorrect responses, such as a flight company that requires printed boarding passes for boarding as an internal rule. These errors can only be considered when the information regarding the particular rules is available in the

previous documentation; however, when generating the response, the chatbot returned something that was potentially correct, but incorrect for the context of that specific company. This distinction is important for better visualization of how to model the domain ontology that must be applied to attempt to constrain response generation.

- Mistaken response repetition - After giving a mistaken response of any of the previous error types, if the chatbot was pressed by the user with the fact that his response was ineffective or mistaken in dealing with the problem at hand, and afterwards the chatbot kept responding with the same mistaken proposed solution, this was classified as a new standalone error.

Domain experts reference response creation, which follows immediately after the error classification, involves a series of interviews with domain experts to formulate what would be considered ideal responses. Notably, the role of human specialists at this point of the pipeline is critical, as domain and operational constraints may not be explicitly formalized in any digital artifact. Therefore, the domain expert responses serve as input to the Ontology Engineering and to the Response Evaluation components.

Essentially, the ontology must contain all entities described in the reference responses, and the response quality must be assessed regarding completeness and correctness when compared to the reference response. The reference response is used to evaluate how well a given algorithm can navigate the knowledge structure database, extracting the relevant entities to generate the prompt that, in turn, creates the chatbot response.

The next component is ontology engineering and verbalization. It is crucial for ontology engineering that the artifact created is compliant with the syntactical rules of the foundational ontology language, thus adhering to the reference WFO; otherwise, the semantic inference would not perform appropriately. Ontology verbalization is the component of transforming the formal axioms of an ontology into natural language statements that can be easily interpreted by human readers, especially domain experts who may lack expertise in formal logic. The goal is to make the implicit meaning of logical constructs accessible and to support the validation of the modeled knowledge. In essence, ontology verbalization combines formal precision with linguistic accessibility, transforming complex logical definitions into coherent and human-readable text fragments (Ellampallil Venugopal and Kumar 2022). It is paramount that the fragments of text adequately represent all relations of a given entity, relating the stereotype of the other entities and of their relation. This way, an effective and reliable description can be created as reference input for the prompt generation, which guarantees access to the immediate context that each relevant entity carries.

Conversation partitioning consists of separating the initial interactions of the conversation, carrying the contexts up to the last moment before the final response of the chatbot, and addressing the problem. The partitioned conversations exclude the final bot's real-life response and transmit the rest of the messages to the entity linking component.

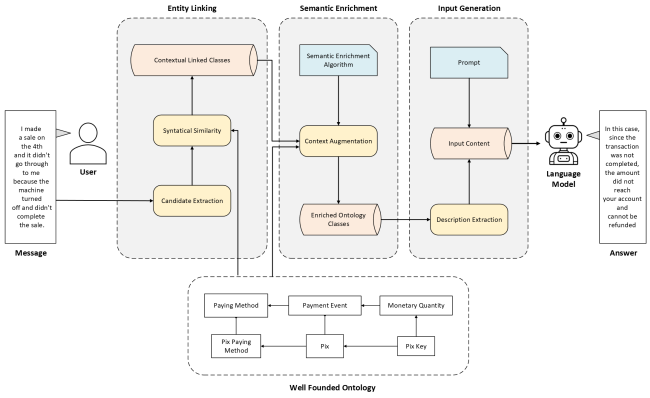


Figure 4. Semantic enrichment component

Figure 4 presents the semantic enrichment component and its interaction with the surrounding framework components in greater detail. A user message is first processed by the entity linking module, which applies NLP preprocessing techniques such as text normalization, tokenization, stop-word removal, and lemmatization to identify candidate entities. These entities are compared against the classes of the foundational ontology using the Jaro–Winkler similarity metric, selected due to its robustness in handling short textual spans, misspellings, abbreviations, and lexical similarities commonly found in informal user-generated messages Cohen et al. (2003). Furthermore, the framework adopts a lightweight symbolic NLP approach instead of complex neural-network-based models to improve explainability, simplicity, and transparency in the entity linking process. Candidate entities whose similarity scores exceed a predefined threshold are selected as matches, resulting in a set of Contextual Linked Classes associated with the Well-Founded Ontology.

The Semantic enrichment is the core of the proposal and is performed automatically. We developed various algorithms to navigate the ontology and extract the ontology fragment that is pertinent to response generation. The central hypothesis is that the algorithms will provide some improvement over the response quality as the descriptions of the classes of the ontology fragment are collected correctly. The ontology must be well-founded so that navigation by the deep semantic algorithm can be conducted, providing context augmentation by the natures of the classes and their relations expressed in their stereotypes. In the example of Figure 4, we show an illustrative snippet of the ontology talking about Pix, the Brazilian central-bank-supported transaction method, to properly define these concepts unique to the particular context. OntoPix (Ferreira et al. 2024), a well-founded conceptual modeling in the Brazilian financial domain, was reused and further expanded upon *. Therefore, based on the algorithm selected, the context augmentation component changes, creating a new set of Enriched Ontology Classes, from which the verbalizations stored as their description are then extracted, thus defining the input content added to the prompt to produce the response.

*The ontology specification is available at https://github.com/fernandabaiao/ontopix-ontology

Finally, with the ontology classes that should be embedded in the context augmentation defined, Prompt Generation occurs straightforwardly, generating a text composed of 4 parts: (i) an initial standard sentence as a role-playing specification for the LLM; (ii) the interactions between the user and the virtual assistant that are part of the conversation up to the point of the LLM enactment; (iii) instructions for response generation, and (iv) the verbalized ontology. A complete example of the execution of all components is provided in the next Section.

Experiment Design

Aiming to evaluate the proposal, a series of algorithms was devised to use the structure of the knowledge graph in different patterns. These techniques, based solely on the graph structure, are the methods entitled light semantics, of which there are seven. All algorithms take as input the return of the Entity Linking (EL) described in the sequel; hence, the entities that they consume are the same, based on the previous messages of the conversation. It is important to note that, since the experiment was conducted utilizing data from a Brazilian company, the original messages are all in Portuguese (the native language of the chatbot's clients).

Entity linking adopts a lightweight, symbolic, and ontology-driven approach for semantic enrichment in financial-domain conversational data. Instead of relying on supervised Named Entity Recognition (NER) architectures or deep neural models (Shen et al. 2015), the proposed method applies deterministic NLP pre-processing combined with approximate string matching to improve explainability, transparency, and computational efficiency.

User messages are initially preprocessed using standard NLP procedures implemented through NLTK and spaCy (pt_core_news_sm). These procedures include lowercasing; text normalization; tokenization; stop-word removal; accent normalization; punctuation and emoji filtering; and lemmatization.

The resulting tokens are then compared against the labels and synonyms defined in the foundational ontology. To evaluate the most suitable similarity metric for the entity linking task, we performed an empirical comparison using representative conversational samples from the financial domain. The evaluated metrics were Jaro–Winkler, Levenshtein Distance, N-Gram similarity, Cosine similarity, Jaccard similarity, Sørensen–Dice and Longest Common Subsequence (LCS).

Each metric was evaluated under different similarity thresholds, using the F1-score as the primary evaluation measure. Figure 5 presents the average F1-score obtained across all evaluated conversations for each similarity metric and threshold configuration.

As illustrated in Figure 5, Jaro–Winkler achieved the best overall balance between precision and recall for lower threshold ranges, outperforming cosine similarity and set-based approaches in handling short textual spans, misspellings, abbreviations, and lexical variations commonly found in informal user-generated messages. Based on these preliminary experiments, Jaro–Winkler was selected as the similarity metric adopted in the entity linking component.

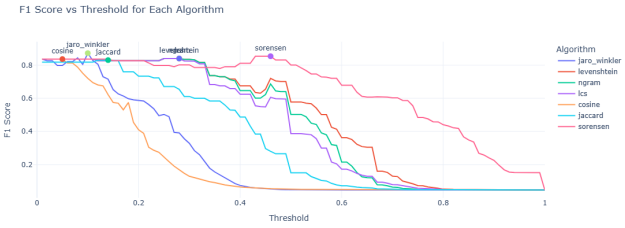


Figure 5. Average F1-score obtained across all evaluated conversations for different similarity metrics under varying threshold configurations.

Candidate ontology entities are selected when the Jaro–Winkler distance is lower than or equal to 0.10 (corresponding approximately to a similarity score of 0.90 or higher). This threshold was empirically defined according to the best observed trade-off between false positives and false negatives during preliminary experiments.

Ambiguity handling is performed through threshold-based filtering and ontology normalization. When multiple ontology entities satisfy the similarity threshold, the framework returns the entities with the highest similarity scores. In the current version, no learned disambiguation model or contextual ranking mechanism is employed, making the proposed approach a simple symbolic baseline for ontology-based semantic enrichment.

Although the proposed strategy has limited semantic generalization compared to embedding-based or neural architectures, it provides important advantages in terms of interpretability, traceability, simplicity, and low computational cost.

Light Semantics

There are seven light algorithms:

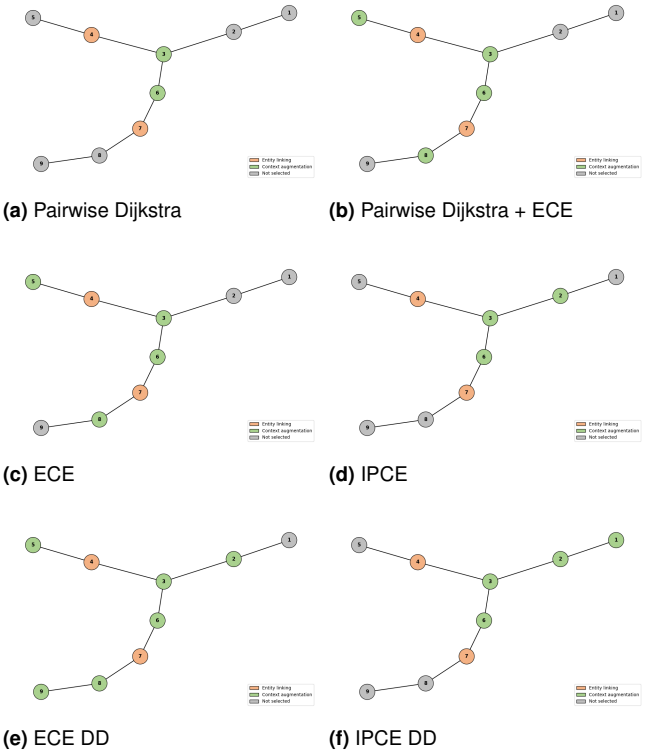
1. Pairwise Dijkstra (PDijk)
2. In-Path Context Expansion (IPCE)
3. In-Path Context Expansion Double Dive (IPCE DD)
4. Entity Context Expansion (ECE)
5. Entity Context Expansion Double Dive (ECE DD)
6. PDijk + ECE
7. Problem-Based

The PDijk algorithm follows the premise that for every pair of entities present in the Contextual Linked Classes list, the shortest path among them, collecting the entities in between, is sufficient to provide the context augmentation. The IPCE algorithm is an expansion of the PDijk that also collects all the entities directly linked to the path formed by the pair of entities extracted by the entity linking. The ECE algorithm considers all entities linked to the Contextual Linked Classes and uses them in the augmentation.

There are also the double dive variants; these algorithms apply the same rule on the context augmented extracted entities, for instance: The IPCE DD adds all entities linked to the path between the entities extracted by the entity linking. The rationale behind it is that providing more context on the connection among entities would facilitate the explanation of the entities. ECE collects all entities adjacent to those extracted by the entity linking. Yet, its Double Dive

counterpart, ECE DD applies the same logic again, collecting the entities linked to the ones previously extracted by the ECE context augmentation. Its rationale is predicated upon gaining more context on what surrounds the entities extracted by the entity linking, therefore understanding them better.

The last graph-based algorithm combines PDijk with ECE, i.e., it applies the PDijk algorithm that collects all entities in the shortest path, followed by the ECE algorithm that collects all entities adjacent to the entities obtained from context extraction. Figure 6 shows an illustrative set of examples to better convey how the algorithms work.



(a) Pairwise Dijkstra **(b)** Pairwise Dijkstra + ECE

(c) ECE **(d)** IPCE

(e) ECE DD **(f)** IPCE DD

Figure 6. Illustrative example of graph-based light semantics

The last light semantics approach is the Problem-Based algorithm. This strategy was envisioned by the pattern of typical problems that emerge naturally in the context. Domain experts were consulted and grouped subsets of the ontology as the ideal set of entities that should be contained in the ideal response for a given typical recurrent set problem.

The Well-founded ontology was split into nine subsets of entities, and new classes were added since the subset is being interpreted as a typical problem, and therefore a new class, Problem, a SITUATION (Costa et al. 2006) under the Unified Foundational Ontology definition, thus describing a given possible state of the world, these situations may trigger an INTERVENTION class, an EVENT as defined by Guizzardi et al. (2013), The occurrence of this intervention may bring about a new SITUATION, Solved problem. Therefore, by the inclusion of these classes and the participation of all relevant entities in the intervention event, the subsets of the ontology can be explicitly modeled in a troubleshooting framework.

The strategy of the algorithm is relatively simple; every subset of the ontology that contains all entities extracted by the entity linking is selected to be used in the prompt generation for the response. The technique proved to be effective; however, it is only functional once a typical pattern of the problem is identified and the domain experts agree with the reference solution. Therefore, it would not be applicable under the real-time need of constructing a response for a novel problem, and therefore, is not necessarily scalable. Another potential issue that may arise is the risk of extracting far more context than needed, as the identified entities at the entity linking component may be common and therefore belong to multiple unrelated problem subsets of the ontology.

This algorithm is vulnerable to hindsight bias, as defined by Kahneman (2012), which is a cognitive bias that emerges when information is selectively recalled, relaying the perception that the response was originally intuitive, when it was actually unknown at the present time. This occurs since the sets of the problem-based ontology fragments that contain the information can only be formed after various incorrect responses are experienced. This means that direct comparison to the other light semantics algorithms would be unfair, as it has the benefit of retroactive expert knowledge to split the sets of entities of interest. This approach contains a final regard that must be analyzed; it is not standardized, despite all domain expert interviewees reaching a consensus for this particular experiment. This may not always be the case, and future alterations in the ontological structure would impact the correctness of the algorithm, as new problem-based sets of ontology fragments would need to be collected. The correctness of the algorithm should be interpreted as a domain-expert-assisted entity selection system, something considerably more complex than the other light semantics methods; however, not entirely semantically aware.

Deep Semantics

The Deep Semantics approach consists of the direct application of the FOPs and Domain Related Ontology Patterns (DROPs) to the entities found by the Entity Linking component. If any of the entities belong to a FOP (presented in the Subsection), a set of semantically relevant entities can be extracted. When the descriptions of the relevant entities are injected into the prompt, the generated response is semantically aware due to absorbing the semantically relevant entities extrapolated from the OntoUML patterns. Therefore, the response-generating system evolves from a merely Neuronal AI system to a Neuro-Symbolic AI System.

Algorithm 1 presents the Deep Semantic Enrichment pseudo-code. Given the set S of seed entities returned by the Entity Linking component, the algorithm computes the semantic closure of S under the Foundational and Domain-Related Ontology Patterns (Falbo et al. 2013). It maintains a worklist W of entities still to be expanded, initialized with S, and an accumulator E^* of entities already processed. At each iteration, an entity e is removed from W; if it has already been processed, it is ignored, which guarantees termination and prevents cycles among mutually dependent classes. Otherwise, two complementary expansions are applied. First, the identity principle of e is resolved by GENERALIZETOKIND, which collects the specialization chain of e up to the ultimate KIND that grants its principle of identity; this is well defined because, in OntoUML, every SORTAL ultimately specializes a unique substance sortal, a KIND being the only terminal symbol of the OntoUML

Table 1. Entities extracted by APPLYPATTERN for each entity e according to its OntoUML stereotype

Stereotype of e	Entities extracted by APPLYPATTERN$(\sigma(e), e, \mathcal{O})$
KIND	$\emptyset$ (terminal symbol; identity principle already resolved)
SUBKIND	ultimate KIND of e
PHASE	sibling PHASEs of the same partition; ultimate KIND
ROLE	mediating RELATOR(s); other ROLEs mediated by them; ultimate KIND
RELATOR	all entities mediated by e
MODE	bearer of e (inherence); external-dependence sources
QUALITY	entity characterized by e
COLLECTIVE	members of e; ultimate KIND of each member
CATEGORY	disjoint rigid sortals abstracted by e
ROLEMIXIN	non-disjoint sortals/roles abstracted by e
EVENT	participants; triggering and brought-about SITUATIONS; created/terminated entities; entities e is historically or externally dependent on (and their dependents)

Algorithm 1: Deep Semantic Enrichment

Input : list of entities S; well-founded ontology $\mathcal{O}$
Output: semantically enriched entity set E^*
$E^* \leftarrow \emptyset$;
$W \leftarrow S$; `// worklist: entities still to be expanded`
while $W \neq \emptyset$ **do**
  remove an entity e from W;
  if $e \notin E^*$ **then**
    $E^* \leftarrow E^* \cup \{e\}$;
    $G \leftarrow$ `GeneralizeToKind(`$e, \mathcal{O}$`)` ;
    `// identity principle: chain up to the ultimate KIND`
    $R \leftarrow$ `ApplyPattern(`$\sigma(e), e, \mathcal{O}$`)` ; `// `$\sigma(e)$`: OntoUML stereotype of` e
    $W \leftarrow W \cup \big((G \cup R) \setminus E^*\big)$;
`Return(`E^*`)`;

grammar (Ruy et al. 2017). Second, APPLYPATTERN applies the pattern associated with the stereotype $\sigma(e)$ of e — where $\sigma(e)$ denotes the OntoUML stereotype of e — returning the entities semantically entailed by that pattern, as summarized in Table 1.

Every newly discovered entity, from either expansion, that has not yet been processed is inserted into W, so the patterns are applied recursively until no new entity is produced, that is, until the fixpoint (the semantic closure of S) is reached. The example below illustrates this recursion: Account Validation, discovered while expanding the Account Activation EVENT, is itself an EVENT and is therefore re-expanded by the EVENT pattern — a closure that a single, non-recursive pass over the entity-linking output could not produce.

For illustration purposes, suppose the following scenario. If the chatbot receives a message, "I sent the required documents; however, the account activation was not approved. Can you help me?" The entities retrieved by the entity linking are: ["Required Documents", "Account Activation"]. Figure 7 shows the relevant ontology fragment (translated to English).

Therefore, after applying the patterns, the algorithm returned the following classes for these reasons: Account Activation is an EVENT, and therefore, all entities that participate in it, are created or terminated by it, must be included, extracting ["Active Account", "Inactive Account"]. As Active Account and Inactive Account are PHASES, after applying the PHASE pattern, their ultimate KIND is extracted, obtaining the ["Account"] entity. Since ACCOUNT ACTIVATION is an EVENT historically dependent on the Account Validation EVENT, the latter is also collected.

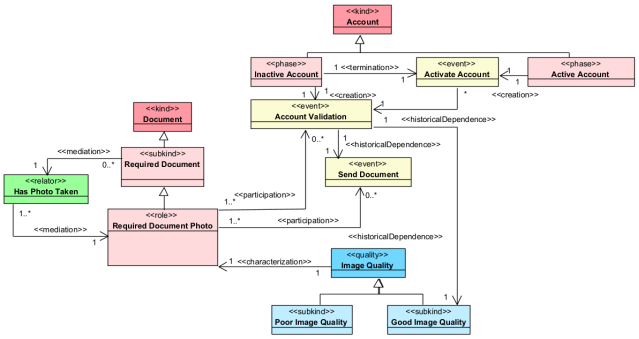


Figure 7. Ontology Fragment

From ACCOUNT VALIDATION, the EVENT pattern is applied again, collecting: ["Required Document Photo"], that being a ROLE has its pattern applied, collecting: ["Has Photo Taken", "Required Document", "Document"]. ACCOUNT VALIDATION is also historically dependent on ["Send Document", "Good Image Quality"]. As GOOD IMAGE QUALITY is the SUBKIND of IMAGE QUALITY, this entity is also collected, and since it is a QUALITY, the entity it characterizes must also be collected, closing the loop on REQUIRED DOCUMENT PHOTOS

With the entities collected, their descriptions are extracted from the verbalization of the ontology, allowing for the creation of the standard prompt that is structured as follows: "You are a chatbot of a credit card machine financial company, respond to the following conversation <<Message>> with the following content: <<Entities' Descriptions>>. This enables the chatbot to generate responses such as: "THE ACCOUNT ACTIVATION depends on the ACCOUNT VALIDATION, which depends on the GOOD IMAGE QUALITY of the REQUIRED DOCUMENT PHOTO. Therefore, either the photo taken does not contain the correct REQUIRED DOCUMENT or the IMAGE QUALITY of the REQUIRED DOCUMENT PHOTO is not good. Please try taking another picture or checking the one you sent."

A dataset was provided containing 60 conversations representative of the most relevant occurrences of errors when generating the responses in the past. The conversations without any error explanation were discarded, as there was not enough information to create the ontology, reducing the dataset to 45 potential conversations; 10 out of 45 were occurrences of complex errors demanding various domains to generate responses, and those were discarded as well. Leaving 35 potential conversations; however, some of the conversations were extremely similar, with typically two or three other instances that were virtually identical,

therefore allowing the experiment to be conducted with 13 conversations, each having a six-digit identification number.

The ontology created based on these conversations comprised 101 classes and 258 associations, which were later shown to experts of the partner company and approved. The approved ontology was then used in a follow-up interview to define what would be the ideal responses for the 13 conversation sessions selected and the sets of entities that were expected to appear in the ideal responses. This creates a reference point that allows evaluation of the algorithms as tools for collecting relevant entities and assessing whether the responses generated by their prompts are correct. After the team evaluated the responses, the team of the partner company was called again to verify whether the evaluation was valid, and they were approved unanimously. To mitigate the stochastic nature of the LLM and improve robustness, each algorithm was run 30 times per conversation, totaling $8 \times 13 \times 30 = 3{,}120$ generated responses over which correctness is computed. The LLM used was ChatGPT 4.0.

End-to-end pipeline example

This subsection presents an end-to-end application of the entire pipeline in a real conversation (session 138184) in the set of real (anonymized) conversations from a financial organization in Brazil. The evaluation of all conversations is presented in more detail in the next Section.

Data gathering

For the data gathering, we used the conversation number 138184, composed of the following interactions:

> *(User) Good morning;*
> *(Chatbot) Good morning, how can I assist you?;*
> *(User) I made a Pix to an unknown person. Can I see his Pix Key?;*
> *(Chatbot) Do you want to know how to view the Pix Key, or do you want to try to make direct contact? Please specify: 1) I want the Key to attempt direct contact, or 2) Other reasons;*
> *(User) 1;*
> *(Chatbot) For safety reasons, I cannot share the Pix Key of other users, and so, it is impossible to make contact. Did this response assist you in any way?*

When the chatbot gives what is considered a ”final response”, that is, an answer that aims to resolve the problem, it always adds the phrase ”Did this response assist you in any way?” Thus, it is always known which parts of the conversation should be used as context (for this example, from the first user message up to his disambiguation numerical response). As a reminder, as noted in previous sections, Pix is an instantaneous payment method created and operated by the Brazilian government.

Error classification

The error classification was performed manually by company experts. This error classification analysis is essential for understanding the business rules and is a key component for understanding the context and information to consider in the ontology engineering process. Note that once this step is done a few times, all relevant domains may be considered and modeled, and therefore, once the ontology is mapped to the specific domain of a subject, this step becomes obsolete.

The error classes obtained after careful discussion with company experts were: Problem nature comprehension; Hallucination; Complex errors; Operational rules; and Mistaken response repetition.

For this particular conversation, two errors were identified: Operational rules and Hallucination. The chatbot does not comprehend that the operational rule regarding sharing Pix keys is only applicable to arbitrary users, since there was a transaction before, the Pix Key is registered on the client’s device by means of the Pix interface. Hallucination is due to the impossibility of contact, as the available Pix Key may be a cellphone number or an e-mail, which may be used as a contact means, if needed. This process is essential, also, to determine whether the particular error is within the scope of the modeling task, as complex errors, for instance, are cases that would require complex reasoning in various domains. An example of such a case would be if a customer were to request an analysis regarding an accident on a given highway, and how it may impact the delivery process of the company, regarding an item ordered.

Domain expert reference response creation

We consulted domain experts and asked them to share the ideal response for this session. This is a manual process; most responses are within a similar domain or have at least some level of overlap regarding the relevant entities. This means that, despite being manual, it is expected that the need for it is reduced as the domain is better captured by the ontology. They gave us the following reference response: ”To consult the Pix Key of a Pix transaction, done to a beneficiary, the client should check the transaction records in the app by accessing the Pix interface in the app interface”. Based on this expert response and the error classification, the ontology fragment was modeled to properly answer this question. It is by the understanding of the errors typically committed and expert-supported ideal responses that the relevant can be fully grasped, and then be modeled in the ontology engineering process

Ontology engineering and verbalization

The development of the ontology is also a manual process; it must be done only once for every domain knowledge fragment, assuming static logic in the business rules over time. Figure 8 shows a fragment of the OntoPIX ontology used in our experimental scenario:

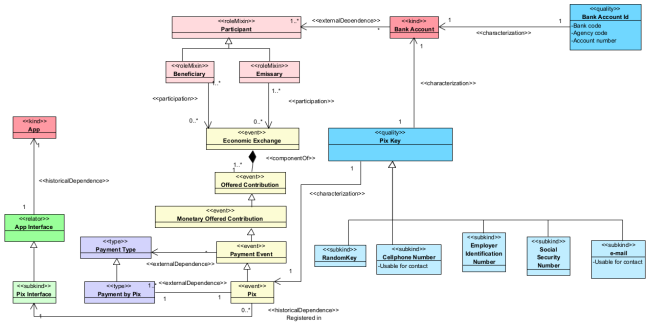


Figure 8. OntoPIX Ontology Fragment

The ontology fragment displays the information present in the previous steps and can be verbalized by declaring the class, its attributes, and its relations, for this particular fragment:

> *"App has a historical dependence on the App Interface, Pix Interface is a specialization of App Interface, Pix Is Registered In Pix Interface, Pix is characterized by a Pix Key, Pix is externally dependent on Payment by Pix, Payment by Pix is a specialization of Payment Type, Pix is a specialization of Payment Event, Payment Event is externally dependent on Payment type, Payment Event is a specialization of Monetary Offered Contribution, Monetary Offered Contribution is a specialization of Offered Contribution, an Offered Contribution is a component of an Economic Exchange, an Economic Exchange has the participation of a Beneficiary and an Emissary, Beneficiary is a specialization of Participant, Emissary is a specialization of Participant, Participant is externally dependent on a Bank Account, a Bank is Characterized by a Bank Account Id, a Bank Account Id has the following attributes: Bank Code, Agency Code, Account Number; A Bank Account is characterized by a Pix Key, a Pix Key is specialized into Random key, Pix Key can be specialized into cellphone number, cellphone number has the attribute usable for contact, Pix Key is specialized into Employer Number Identification, Pix Key is specialized into Social Security Number, Pix Key is specialized into e-mail, e-mail has the attribute usable for contact."*

Worth mentioning, the complete ontology comprises 101 classes, of which only 20 were reused from existing ontologies, while the remaining 81 classes were specifically designed to represent the operational concepts, business rules, and domain-specific constraints of the particular domain of discourse being modeled. This modeling process involved capturing organizational procedures and relationships that are not adequately represented by general-purpose financial ontologies, resulting in a knowledge structure tailored to the decision-making requirements of the application domain.

The overall contribution of the proposed framework is situated within the Neuro-Symbolic AI paradigm by combining knowledge graphs for contextual storage with ontological reasoning grounded in structural patterns that are ontologically well-founded in UFO for context engineering. The core novelty of the proposed framework lies in the Deep Semantic Enrichment component, which leverages the well-founded semantic nature of ontological entities, rather than solely their graph-like structure, to support more precise context extraction and thus cost-effective, yet complete context engineering for further LLM interactions. This is realized through the semantic enrichment and relevant entity expansion mechanism that exploits the inherent structure of FOPs and DROPs in OntoUML, grounded in UFO, a technique whose implementation and impact on LLM performance constitutes the primary contribution of this work.

The verbalization process is entirely automatic and is done with the JSON of the ontology. After, the verbalization is stored in a table linking the class and its verbalization, considering the ontology. Depending on the entities detected in the Entity linking step, a different sequencing of the verbalization is concluded; the same is true for the Semantic enrichment step. This means that the exact verbalization is dependent on the combination of the classes detected by the Entity linking and the classes detected in the Semantic enrichment; the changes in this sequence may affect the LLM's attention, as information that may be relevant is moved to different parts of the Prompt generation step. Therefore, the response evaluation must consider correctness to validate the Semantic enrichment as a response generation strategy and an evaluation regarding the entities extracted against the entities of the ideal response, to see how well the algorithm navigates the ontology in the attempt to find relevant information and not extract unnecessary information.

Conversation partitioning

This is one of the most straightforward steps; we simply must split the conversation between the chatbot's previous interactions and its final response to simulate response performance with semantic enrichment. This is done by tracking the final response of the chatbot, always indicated by the phrase "Did this response assist you in any way?" Nevertheless, it is possible that for a given conversation, the response was insufficient, eliciting a second response; these, and other rare cases in which the chatbot committed the same mistake twice (which received a Mistaken response repetition error classification). Because of these edge cases, the conversation partitioning was done in the following manner:

The conversation partitioning step consists of finding the first instance of a chatbot response with the string "Did this response assist you in any way?" All messages before this moment are added to the "previous context", the string used for the entity linking and context, and the rest of the conversation is discarded. Therefore, the previous context consists of " User- Good morning. Chatbot- Good morning, how can I assist you? User- I made a Pix to an unknown person. Can I see his Pix Key? Chatbot- Do you want to know how to view the Pix Key, or do you want to try to make direct contact? Please specify: 1) I want the Key to attempt direct contact, or 2) Other reasons. User: 1. Chatbot- For safety reasons, I cannot share the Pix Key of other users, and so, it is impossible to make contact. Did this response assist you in any way?" was discarded.

Entity linking

The Entity linking analyzes the previous context to find matching terms in the ontology; in this case, the terms found were "Pix" and "Pix Key", which were the terms used for the Semantic enrichment.

Semantic Enrichment

For this part of the end-to-end pipeline example, we focused solely on the deep semantic enrichment, as the light semantic algorithms were explained in depth in the previous section and properly demonstrating the sequencing on the real ontology (that is considerably bigger than the example of the Light Semantics session, and the ontology fragment formed by the ideal response for this session) would take

a considerable time and the visualization for some of the algorithms, specially the ones that operate under the double dive logic, would be confusing. The Deep semantic enrichment process consists of applying deep semantics, by means of FOPs and DROPs present in the structure of the ontology, to determine relevant information from the given set of entities provided by the previous step, [”Pix”, ”Pix Key”]. This process is entirely automatic by manipulating the data of the JSON that represents the ontology.

The deep semantics capitalizes on the deeper semantic structure of the well-founded ontology and its structural patterns to determine what entities are relevant for a given set of entities, due to their nature and relations. Pix is a specialization of some other entity; therefore, we must generalize it back until the entity that grants its identity principle is detected. So the entities ”Payment Event”, ”Monetary Offered Contribution”, and ”Offered Contribution” must be collected. As they are generalizations, any pattern-based relations they possess should also be relevant to ”Pix”, which leads the algorithm to determine that ”Payment Type” is important, due to the external dependence between it and Monetary Offered Contribution. The next relevant relation is the participational composition between Offered Contributions and Economic Exchange events. ”Economic Exchange” is, therefore, a relevant entity, and as it is an event, the event FOP is applied to it; therefore, all classes that participate in it are externally or historically dependent on it, or that it is externally or historically dependent on, along with entities terminated, brought about, created, or triggered by it would be relevant. This means that the entities ”Beneficiary” and ”Emissary” are relevant, and as they are both specializations of ”Participant”, all of them were included. Pix is also an event; the event pattern is applied to it as well, capturing the entities ”Pix Interface” and ”Payment by Pix”, both of which are specializations; however, the specialization of Payment by Pix was already included, so only ”App Interface” is added. Pix is also characterized by Pix Key; therefore, ”Pix Key” is relevant, but it is already considered in the initial set. No new entities are extracted from Pix Key.

This provided the following semantically enriched set of entities: [”Pix”, ”Pix Key”, ”Payment Event”, ”Monetary Offered Contribution”, ”Offered Contribution”, ”Payment Type”, ”Economic Exchange”, ”Beneficiary”, ”Emissary”, ”Participant”, ”Pix Interface”, ”Payment by Pix”, ”App Interface”]. The verbalization of this set of entities was used in the prompt generation. Note that the new entities extracted by the FOPs and DROPs processes themselves were [”Beneficiary”, ”Emissary”, ”Payment by Pix”, ”Pix Interface”]; all other entities derive from specializations.

Prompt generation

The prompt generation process produces a text composed of four components: a role-playing specification, the conversation context, the response instructions, and the verbalized ontology. The role-playing specification consists of a standard introductory statement:

> *”Consider that you are playing the role of a customer service chatbot to a Brazilian financial institution that operates the systems of credit card machines. Consider the conversation below and generate a response:”*

The conversation context follows with all the interactions exchanged between the chatbot and the user before enacting the LLM:

> *(User) Good morning;*
>
> *(Chatbot) Good morning, how can I assist you?*
>
> *(User) I made a Pix to an unknown person. Can I see his Pix Key?*
>
> *(Chatbot) Do you want to know how to view the Pix Key, or do you want to try to make direct contact? Please specify: 1) I want the Key to attempt direct contact, or 2) Other reasons.*
>
> *(User) 1*

The response instruction is the following standard sentence:

> *”For the response, add the phrase ’Did this response assist you in any way?’, consider the information provided below:”*

Finally, the verbalization of the ontology fragment follows:

> *”Pix Is Registered In Pix Interface, Pix is characterized by a Pix Key, Pix is externally dependent on Payment by Pix, Payment by Pix is a specialization of Payment Type, Pix is a specialization of Payment Event, Payment Event is externally dependent on Payment type, Payment Event is a specialization of Monetary Offered Contribution, Monetary Offered Contribution is a specialization of Offered Contribution, an Offered Contribution is a component of an Economic Exchange, an Economic Exchange has the participation of a Beneficiary and an Emissary, Beneficiary is a specialization of Participant, Emissary is a specialization of Participant, Participant is externally dependent on a Bank Account. Pix Interface is a specialization of App Interface.”*

The complete generated prompt is then sent to the LLM, which, in this real example scenario, returned the following response:

> *”We do not inform the* PIX KEY *of other clients through this channel. A* PIX PAYMENT EVENT *involves an* EMISSARY *and a* BENEFICIARY *as* PARTICIPANTS*; however, the* PIX *is registered in the* PIX INTERFACE *of your* APP INTERFACE*. You may seek the* PIX KEY *there. If you have any difficulty finding it, let me know, and I will signal a human member of the support team to further assist you. Did this response assist you in any way?”*

The next step is the response evaluation, which is described in the following Section together with the evaluation of all conversations considered in the experimental evaluation of our proposal. The evaluation must be done collectively, as shown in the next session, since some of the evaluation metrics are done by comparing the various semantic enrichment strategies.

Results

The performance of the light semantics algorithms and the deep semantics algorithm must be evaluated in two dimensions: i) The quality of the algorithm as an inference machine to extract relevant entities, i.e., from the starting point that is the set of entities derived from entity linking, the algorithm must select the entities relevant to generate the response ii) The quality of the responses generated by the algorithm. Both dimensions must be evaluated separately, as the neural component of the Neuro-Symbolic AI chatbot may still be able to produce a correct response from the inferences of an incomplete set of entities. Similarly, the set of entities selected to have their descriptions injected into the prompt's context may influence the likelihood of the LLM to generate hallucinations in the response.

Therefore, some metrics were derived to measure the capabilities of the algorithms in both regards. The number of entities extracted was analyzed, and three metrics were derived from it:

i) Completeness: evaluates the entities extracted, and what percentage of them belong to the reference set of entities assigned to each session by the domain experts as the appropriate entities to be used in the generation of the ideal reference response;

ii) Excessive Information Score: punishes the extraction of unnecessary entities. It is calculated by dividing the number of entities extracted by the algorithm that do not belong to the ideal set of entities by the number of entities that belong to the ideal set. This value is then normalized among all algorithms for a given question, applying the min-max technique. Therefore, the metric is highly sensitive to the number of entities of the ideal set, punishing more severely cases in which few entities would be required to generate a response. This metric, therefore, ranges from 0% to 100%, with 100% corresponding to the worst-performing algorithm. This metric is particularly useful for comparing the semantic enrichment algorithms because, although it is not directly used to control prompt length, it is proportional to it. Consequently, it provides a valuable proxy for prompt complexity and the amount of semantic information incorporated by each approach.

iii) Correctness: measures the percentage of correct responses generated by the prompt of each algorithm for a given conversation session. Thirty responses are generated for each algorithm and session to account for the stochasticity of the LLM. The responses are considered correct if they match the ideal reference response approved by the domain experts. Correctness was measured manually by the total consensus of a group of experts specifically designated for this task. Each expert manually and blindly evaluated the responses with no information about the algorithm used to generate the answer, having access only to the reference response, the previous conversational context, and the generated responses.

Table 2 shows the number of entities extracted by each algorithm.

Some observations can be derived from Table 2. The Dijkstra-based algorithms (Pairwise Dijkstra, In-Path CE, In-Path CE DD, and Dijk + ECE) have various instances where they returned no new entities extracted beyond the ones detected by the entity linking. This occurs whenever only one entity was extracted by the entity linking, since there is no pair of entities, no paths are produced, and no new entities can be added.

Table 2. Number of Extracted Entities

Session/Alg	Pairwise Dijk	In-path CE	ECE	ECE DD	In-path CE DD	Dijk ECE	Problem-Based	Deep
138056	**0**	**0**	2	4	**0**	**0**	**16**	5
157753	**0**	**0**	6	17	**0**	**0**	16	**17**
138184	**0**	**0**	5	**16**	**0**	**0**	14	4
138063	**0**	**0**	5	**16**	**0**	**0**	14	7
142041	**4**	9	10	15	**35**	14	9	7
138035	**4**	6	14	**32**	9	16	24	7
138041	**4**	8	8	**22**	18	11	13	7
138053	**3**	10	10	**33**	16	17	9	4
145816	**5**	15	13	**36**	30	24	22	8
150793	**0**	**0**	3	**7**	**0**	**0**	**0**	**7**
155994	**0**	**0**	2	10	**0**	**0**	**25**	4
156042	**0**	**0**	8	32	**0**	**0**	**42**	2
Average extraction	**1.67**	4.00	7.17	20.00	9.00	6.83	17.00	6.58

The ECE DD algorithm results in the highest average of entities extracted, followed closely by the problem-based algorithm; however, the problem-based algorithm is mostly influenced by session 156042, where only a very common entity was returned by the algorithm. This led to the collection of multiple ontology fragments and an excessive number of entities. The In-Path DD algorithm extracts an excessive number of entities frequently, whenever more than one entity is provided by the entity linking. The Deep algorithm collected a low number of entities, considering that the overall average for entity extraction is approximately 9, which is positive, as the main idea behind it was to limit excessive context injection while giving enough to respond to the questions precisely. The response quality aspect of the algorithms can be described by Tables 3 and 4

Table 3. Percentage of correct responses for each algorithm, best performer of each session marked in green

Session/Alg	Pairwise Dijk	In-path CE	ECE	ECE DD	In-path CE DD	Dijk ECE	Problem-Based	Deep
138056	66%	76%	60%	86%	40%	68%	76%	86%
157753	60%	90%	10%	12%	70%	50%	56%	92%
138184	0%	0%	23%	30%	0%	0%	90%	77%
138063	0%	80%	40%	93%	90%	0%	93%	90%
142041	3%	13%	26%	0%	0%	0%	0%	100%
138035	7%	27%	20%	40%	23%	0%	80%	97%
138041	10%	17%	36%	3%	3%	0%	10%	83%
138053	0%	100%	27%	100%	80%	70%	100%	93%
145816	3%	0%	50%	10%	0%	30%	67%	87%
150793	0%	0%	0%	0%	0%	0%	0%	83%
155994	0%	0%	50%	0%	0%	0%	97%	67%
156042	0%	0%	90%	0%	0%	50%	70%	27%
Average	20%	34%	36%	31%	26%	22%	62%	81%

The graph-based algorithms generate between 20% and 36% of correct responses; this seems low; however, it is important to remember that the sessions explored at the baseline, that is, with no ontological approach (pure LLM), achieved a success rate of at most 3%. The Problem-based algorithm represents a major improvement, reaching 62% correctness, while the Deep algorithm achieves a staggering 81% correctness, which is impressive, considering that the problem-based algorithm has the major advantages of domain expert intervention and hindsight bias, as discussed previously.

From Table 3 we can also see that the Deep algorithm consistently is the most precise algorithm, 58.3% of the time, it is among the three best alternatives, for 11 of the 12 occurrences, equating to 91.6% of the sessions. As a matter of fact, the Problem-based and Deep algorithms are considerably superior to all the other alternatives, being

the best performers 91.6% of the time, when compared to the overall performance of the graph-based light semantics algorithms.

This indicates that the Deep algorithm indeed reduced hallucination by constraining the LLM's reasoning by giving a select set of semantically relevant entities based on their ontological nature. This perception is further detailed in Tables 4 and 5.

Table 4. Completeness percentage for each algorithm, best performer of each session marked in green

Session/Alg	Pairwise Dijk	In-path CE	ECE	ECE DD	In-path CE DD	Dijk ECE	Deep
138056	0%	0%	18%	36%	0%	0%	45%
157753	0%	0%	17%	44%	0%	0%	33%
138184	0%	0%	33%	88%	0%	0%	44%
138063	0%	0%	0%	25%	0%	0%	100%
142041	33%	75%	42%	100%	100%	75%	66%
138035	29%	29%	71%	93%	29%	71%	50%
138041	50%	83%	50%	83%	100%	83%	33%
138053	15%	50%	30%	75%	80%	65%	20%
145816	13%	19%	38%	63%	32%	43%	50%
150793	0%	0%	0%	0%	0%	0%	57%
155994	0%	0%	40%	40%	0%	0%	40%
156042	0%	0%	20%	70%	0%	0%	10%
Average Completeness	12%	21%	30%	60%	28%	28%	46%
Correctness	20%	34%	36%	31%	26%	22%	81%

Considering the way that the Problem-based algorithm was envisioned, it would be unfair to measure its Completeness and Excessive Information Score; therefore, only the graph-based algorithms and the Deep semantic enrichment algorithm were considered for this measurement.

Table 4 shows that there is no apparent correlation between completeness and correctness, as the Deep algorithm has a 46% completeness score and is nevertheless over 2.2 times more precise than the second-best candidate. The highest average completion score belongs to the ECE DD algorithm at 60%. 100% completion occurred only 4 times, three belonging to the double dive light semantics algorithms and one to the Deep semantics algorithm. Table 4

Table 5. Excessive Information Score for each algorithm, worst values of each session marked in red

Session/Alg	Pairwise Dijk	In-path CE	ECE	ECE DD	In-path CE DD	Dijk ECE	Deep
138056	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
157753	0.0%	0.0%	34%	100%	0.0%	0.0%	0.0%
138184	0.0%	0.0%	25.4%	100%	0.0%	0.0%	11.9%
138063	0.0%	0.0%	25.8%	100%	0.0%	0.0%	3.2%
142041	0.0%	0.0%	21.9%	100%	13%	21.9%	0.0%
138035	0.0%	25.2%	20.3%	100%	95.1%	44.8%	0.0%
138041	0.0%	9.5%	18.9%	100%	80.9%	23.8%	14.5%
138053	0.0%	0.0%	22.2%	100%	0.0%	22.2%	0.0%
145816	11.5%	46%	27%	100%	95.7%	65%	0.0%
150793	0.0%	0.0%	43%	100%	0.0%	0.0%	43%
155994	0.0%	0.0%	0.0%	100%	0.0%	0.0%	25%
156042	0.0%	0.0%	22.2%	100%	0.0%	0.0%	0.0%
Average EIS	1.0%	6.7%	21.7%	91.7%	23.7%	14.8%	8.1%
Correctness	20%	34%	36%	31%	26%	22%	81%

As the Excessive Information Score is dependent on the size of the set of entities that belong to the ideal response, there that extremely large values are reached due to a low number of entities in the ideal response, which is particularly noticeable for the ECE DD algorithm, reaching the constantly the maximum normalized value of 100%, being the highest value for all sessions and attaining an average of 91.7%, over three times larger to the second largest Excessive Information Score, belonging to the other double dive algorithm. The Dijkstra-based algorithms have consistently low Excessive Information Scores, mostly due to the occurrences where no new entities are extracted. The Deep algorithm achieved a considerably low Excessive Information Score of 8.1%, placing it in the third lowest place, between 6.7% and 14.8%. This indicates that the Deep algorithm proved to be effective in constraining the injection of undue context. Figures 9 and 10 show the relation between completeness, Excessive Information Score, and correctness.

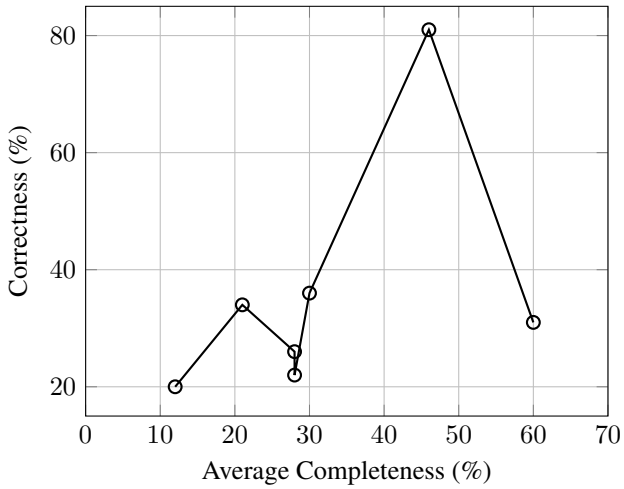


Figure 9. Correctness versus Average Completeness

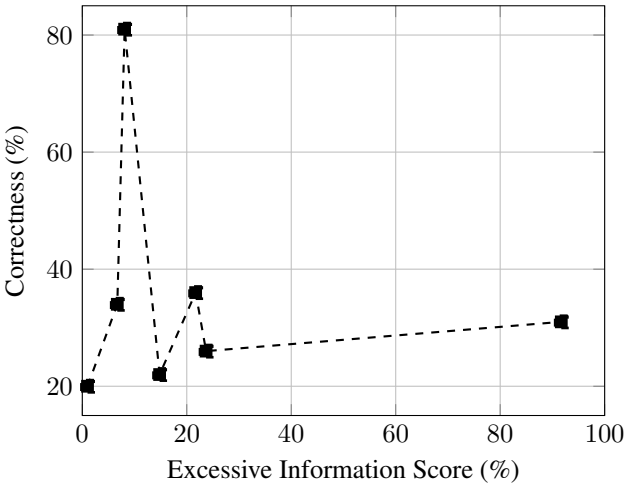


Figure 10. Correctness versus Excessive Information Score

The figures clearly indicate that there is no direct correlation in both completeness and Excessive Information Score to correctness, as the Deep algorithm reaches extremely high correctness despite only having a 46% completeness, showing that the proper selection of entities is sufficient to interact with the LLM and generate correct responses from the constrained inference component of the neural component of the system. The number of extracted entities is directly proportional to prompt length, and by analyzing the results, it can be seen that there is no direct comparison between prompt length and correctness.

Simultaneously, the Deep algorithm reached a low Excessive Information Score; however, other algorithms with lower Excessive Information Scores(EIS) are considerably

worse, while the ECE DD algorithm, with the highest EIS, was not the worst performer.

To assess the contribution of each ontology pattern to the Deep Semantic Enrichment, every entity included in the enriched prompts was traced back to the pattern responsible for its extraction, as shown in Table 6. Taxonomic structures (generalization to the ultimate KIND and SUBKIND/PHASE specializations) account for 38.4% of the contributed entities, reflecting the central role of the identity principle in the algorithm, while the EVENT pattern and historical dependence jointly account for 44.2%, which is consistent with the predominantly processual nature of the domain and with their role in assembling the explanatory chains injected into the prompts. The RELATOR and ROLE patterns contribute marginally in this particular ontology, which contains few relators; their relevance is therefore expected to vary with the modeled domain rather than being negligible in general.

Table 6. Ontology patterns responsible for the entities contributed by the Deep Semantic Enrichment across all sessions (86 entities beyond the entity-linking seeds, matching Table 2; each entity credited to the pattern that extracted it).

Pattern	Entities	Share
Taxonomic structure (generalization to KIND; SUBKIND/PHASE specializations)	33	38.4%
EVENT (participants, created/terminated, triggering)	19	22.1%
Historical dependence	19	22.1%
Parthood (component of)	7	8.1%
QUALITY/MODE (characterization)	6	7.0%
Other (SITUATION, external dependence)	2	2.3%
Total	86	100%

There seems to be an emphasis on the quality of the entities extracted and not their quantity. Either way, the Deep semantic enrichment proved extremely effective, achieving superior performance across all relevant evaluation metrics. The gains of the deeper semantics applied can be perceived by Figures 7 and 8, comparing the average of the light semantics algorithms with the Deep correctness.

Figure 7 shows that the Deep algorithm outperforms in every metric, going from 30% to 46% completeness, reducing the Excessive Information Score from 26.6% to 8.1%, and response correctness increased from 28% to 81%.

Figure 12 shows the percentage difference from the light semantics to the Deep semantics metrics, having a 53% increase in completeness, a 69.5% reduction in the Excessive Information Score, and an increase in response correctness by 189%.

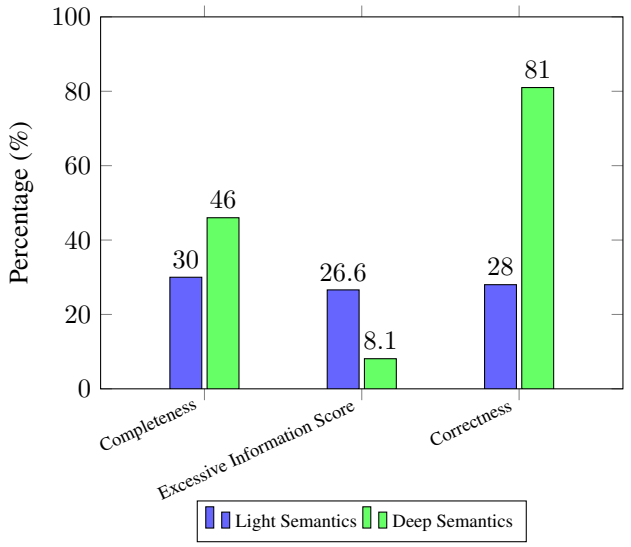


Figure 11. Comparison of completeness, Excessive Information Score, and correctness between Light and Deep Semantics.

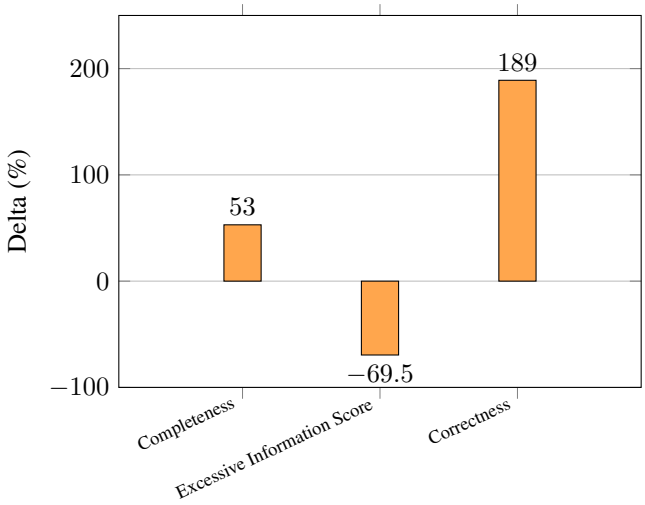


Figure 12. Relative percentage variation of Deep Semantics with respect to Light Semantics.

Conclusion

This paper presented a novel Neuro-Symbolic framework to improve question-answering systems that combines knowledge graphs for contextual storage with ontological reasoning grounded in structural patterns (including foundational ontology patterns and domain-related ontology patterns) which are ontologically well-founded in UFO.

The contribution of the proposed framework includes the proposal and systematic evaluation of the impact of deep semantic enrichment on LLM-based knowledge extraction and response generation within chatbot conversations. From a holistic perspective, the proposal is a novel implementation of the Neuro-Symbolic AI paradigm, combining a heavyweight symbolic representation of domain-specific knowledge grounded in structural patterns that are ontologically well-founded in UFO with the use of connectionist-based LLMs comprising text generation over general knowledge. By leveraging the well-founded semantic nature of ontological entities, rather than solely their graph-like structure (such as KG-based approaches), the proposed framework goes beyond the usual graph-based RAG approaches in supporting more precise reasoning during context engineering for prompt generation. The proposed framework was systematically evaluated using real conversations from a chatbot of a Brazilian company in the financial domain, evidencing the positive impact of the deep semantic enrichment over

a combination of metrics specifically and systematically tailored for the evaluation of such scenarios, addressing response correctness, completeness, and (lack of) excessive information.

The Neuro-symbolic paradigm adopted by our proposal, which uses the ontology as a symbolic and formal representation of domain constraints, provides a structural knowledge layer of rules with reasoning abilities through inference mechanisms that may be performed on top of it, with neither dependence nor additional costs due to the use of tokens when using stochastic LLMs. Moreover, the knowledge structure also represents a valuable asset for the organization, acting as a contract of domain specifications among stakeholders and integrated systems.

The results showed performance improvement of the Deep semantics algorithm in the perspective of entity extraction and response generation when compared to the various light semantics algorithms that make use solely of the graph-like properties of the ontology and the problem-based algorithm formulated in conjunction with the partner company's team of domain experts.

Concerning scalability, we argue that the usage of the proposed solution is scalable in practice. At runtime (that is, when it receives a conversation that it needs to respond to), the components that are effectively executed (namely, Conversation partitioning, Entity Linking, Semantic Enrichment, and Prompt generation) are all automatic, and the entire execution is almost immediate (in our experiments, each conversation took a few seconds to be responded to). Moreover, the execution time is directly proportional to the number of elements (classes and relationships) in the ontology, and, given that our solution requires a type-level ontology, the number of elements is typically low.

An additional consideration regarding scalability concerns the ontology itself. The proposed approach presupposes the existence of a well-founded domain ontology, while its construction, expansion, and maintenance are outside the scope of this work. Nevertheless, according to the collaborating company, a substantial portion of its workforce is currently dedicated to handling operational errors and exceptions. As the proposed solution aims to reduce the occurrence of such errors, the resulting reallocation of human resources could partially offset the effort required for ontology maintenance. Furthermore, maintaining and extending a type-level ontology primarily involves updating business concepts and relationships rather than large volumes of instance data, making these activities manageable with limited additional training for existing personnel. Consequently, the long-term operational costs associated with ontology evolution are expected to be compatible with the anticipated gains in process efficiency and service quality.

Our proposal aligns with the tendency in the literature to integrate knowledge graphs into RAG pipelines. Chen (2025) highlights that structured retrieval enables more robust multi-hop reasoning and reduces hallucinations in settings where flat text retrieval fails. Hence, in our proposed framework, the production baseline at the partner company is a text-centric RAG system that, while for many queries still produces systematic errors in cases requiring structured semantic reasoning. The semantic enrichment interventions described in this paper were specifically designed to address this gap, achieving measurable gains on the error classes where the text-centric baseline falls short.

In its current implementation, the gains observed by the proposed approach are limited to conversations about topics within the scope of the domain ontology. When the user starts a conversation mentioning terms that are out of scope, the Entity Linking component does not match any class of the ontology, and no relevant entities are retrieved, thus preventing any semantic enrichment of the generated prompt. Consequently, the system falls back to a generic response, effectively behaving as the underlying base LLM. For example, when asked "Who was the top scorer of the 1978 World Cup?", both the enriched and baseline systems produced the same response.

Another limitation concerns the extent to which the broader impact of the proposed approach can be assessed. The experimental evaluation was conducted using data provided by the collaborating company, whose availability was inherently limited by operational and confidentiality constraints.

Although the results consistently demonstrated improvements in response quality and expert evaluation metrics within the available dataset, the restricted number and diversity of the data limit the ability to generalize these findings to a wider range of organizational settings and application domains. Future work should, therefore, consider evaluating the approach on larger and more heterogeneous datasets, encompassing additional companies, business processes, and conversation scenarios. Such an expanded evaluation would enable a more comprehensive assessment of the robustness, scalability, and practical impact of ontology-based semantic enrichment in industrial conversational systems. Other Interesting venues for future work will evaluate further error categories by extending the domain ontology automatically and experimenting on the generation of SPARQL queries to be executed on the graph, whose results could then be used to fill the prompt.

Data Availability Statement

All data and artifacts used in the experiments are publicly available, comprising (i) OntoUMLLoader, a dependency-free Python library for loading OntoUML models serialized in the OntoUML JSON Schema into navigable Python objects, which serves as the ontology access layer for the Deep Semantics component; (ii) the experiment materials, including the codification of the ontology in json, entity-linking outputs, light-semantics algorithms, generated responses, and expert evaluation results; and (iii) the domain ontology specification. The repositories are available at https://github.com/lucasmadda/OntoUMLLoader and https://github.com/fernandabaiao/ontopix-ontology.

Acknowledgements

Acknowledgments: The authors thank StoneLab for providing the anonymized data and funding the research. Regarding the use of generative AI: It was only employed as a tool for spelling correction and making adjustments to increase text fluidity; all the text was developed, examined, and approved by the authors.

References

Alberts IL, Mercolli L, Pyka T, Prenosil G, Shi K, Rominger A and Afshar-Oromieh A (2023) Large language models (LLM) and ChatGPT: what will the impact on nuclear medicine be? *European Journal of Nuclear Medicine and Molecular Imaging* 50(6): 1549 – 1552. DOI:10.1007/s00259-023-06172-w. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85149473505&doi=10.1007%2fs00259-023-06172-w&partnerID=40&md5=6588ad1af56fde75598905ab3794b92a. Type: Editorial.

Birkun AA and Gautam A (2023) Large Language Model (LLM)-Powered Chatbots Fail to Generate Guideline-Consistent Content on Resuscitation and May Provide Potentially Harmful Advice. *Prehospital and Disaster Medicine* 38(6): 757–763. DOI:10.1017/S1049023X23006568. URL https://www.cambridge.org/core/product/identifier/S1049023X23006568/type/journal_article.

Cadeddu A, Chessa A, De Leo V, Fenu G, Motta E, Osborne F, Reforgiato Recupero D, Salatino A and Secchi L (2024) A comparative analysis of knowledge injection strategies for large language models in the scholarly domain. *Engineering Applications of Artificial Intelligence* 133: 108166. DOI:10.1016/j.engappai.2024.108166. URL https://linkinghub.elsevier.com/retrieve/pii/S0952197624003245.

Carvalho I and Ivanov S (2024) ChatGPT for tourism: applications, benefits and risks. *Tourism Review* 79(2): 290 – 303. DOI:10.1108/TR-02-2023-0088. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85150600701&doi=10.1108%2fTR-02-2023-0088&partnerID=40&md5=ae2aa3afbb556d477e5f5b278a472459. Type: Article.

Chen R (2025) Retrieval-augmented generation with knowledge graphs: A survey. In: *Submitted to Computer Science Undergradaute Conference 2025 @ XJTU*. URL https://openreview.net/forum?id=ZikTuGY28C. Under review.

Cohen WW, Ravikumar P, Fienberg SE et al. (2003) A comparison of string distance metrics for name-matching tasks. In: *IIWeb*, volume 3. pp. 73–78.

Costa PD, Guizzardi G, A Almeida JP, Pires LF and Sinderen MV (2006) Situations in conceptual modeling of context. In: *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*. pp. 6–6. DOI:10.1109/EDOCW.2006.62.

Deng J and Lin Y (2023) The Benefits and Challenges of ChatGPT: An Overview. *Frontiers in Computing and Intelligent Systems* 2(2): 81–83. DOI:10.54097/fcis.v2i2.4465. URL https://drpress.org/ojs/index.php/fcis/article/view/4465.

Edge D, Trinh H, Cheng N, Bradley J, Chao A, Mody A, Truitt S, Metropolitansky D, Ness RO and Larson J (2024) From Local to Global: A Graph RAG Approach to Query-Focused Summarization. DOI:10.48550/ARXIV.2404.16130. URL https://arxiv.org/abs/2404.16130. Version Number: 2.

Ehrlinger L and Wöß W (2016) Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)* 48(1-4): 2.

Ellampallil Venugopal V and Kumar PS (2022) Verbalizing but Not Just Verbatim Translations of Ontology Axioms. In: Leiva LA, Pruski C, Markovich R, Najjar A and Schommer C (eds.) *Artificial Intelligence and Machine Learning*, volume 1530. Cham: Springer International Publishing. ISBN 978-3-030-93841-3 978-3-030-93842-0, pp. 170–186. DOI:10.1007/978-3-030-93842-0_10. URL https://link.springer.com/10.1007/978-3-030-93842-0_10. Series Title: Communications in Computer and Information Science.

Falbo RA, Guizzardi G, Gangemi A and Presutti V (2013) Ontology patterns: clarifying concepts and terminology. In: *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns - Volume 1188*, WOP'13. Aachen, DEU: CEUR-WS.org, p. 14–26.

Ferguson S and Hebels R (2003) Access to information resources. In: *Computers for Librarians*. Elsevier. ISBN 978-1-876938-60-4, pp. 81–109. DOI:10.1016/B978-1-876938-60-4.50009-4. URL https://linkinghub.elsevier.com/retrieve/pii/B9781876938604500094.

Ferreira G, Peixoto M, Maddalena L and ao FB (2024) Ontopix: Modelagem conceitual preliminar bem fundamentada no domínio financeiro brasileiro. In: *CEUR Workshop Proceedings*, volume 3905. p. –?

Gruber TR (1993) A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2): 199–220. DOI:10.1006/knac.1993.1008. URL https://linkinghub.elsevier.com/retrieve/pii/S1042814383710083.

Guarino N (1998) Some ontological principles for designing upper level lexical resources. URL https://arxiv.org/abs/cmp-lg/9809002.

Guarino N, Oberle D and Staab S (2009) What Is an Ontology? In: Staab S and Studer R (eds.) *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-92673-3, pp. 1–17. DOI:10.1007/978-3-540-92673-3_0. URL https://doi.org/10.1007/978-3-540-92673-3_0.

Guizzardi G (2005) *Ontological foundations for structural conceptual models*. PhD Thesis - Research UT, graduation UT, University of Twente. ISBN: 90-75176-81-3 Issue: 05-74 Series: CTIT PhD Thesis Series.

Guizzardi G and Guarino N (2024) Explanation, semantics, and ontology. *Data & Knowledge Engineering* 153: 102325. DOI:10.1016/j.datak.2024.102325. URL https://linkinghub.elsevier.com/retrieve/pii/S0169023X24000491.

Guizzardi G, Wagner G, De Almeida Falbo R, Guizzardi RSS and Almeida JPA (2013) Towards Ontological Foundations for the Conceptual Modeling of Events. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Ng W, Storey VC and Trujillo JC (eds.) *Conceptual Modeling*, volume 8217. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-41923-2 978-3-642-41924-9, pp. 327–341. DOI:10.1007/978-3-642-41924-9_27. URL http://link.springer.com/10.1007/978-3-642-41924-9_27. Series Title: Lecture Notes in Computer Science.

Kahneman D (2012) *Thinking, fast and slow*. Penguin psychology. London: Penguin Books. ISBN 978-0-14-103357-0.

Lutz C, Seylan I and Wolter F (2012) Mixing open and closed world assumption in ontology-based data access: Non-uniform data complexity. *CEUR Workshop Proceedings* 846: 268–278.

Maedche A and Staab S (2001) Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16: 72–79. DOI:10.1109/5254.920602.

McKenna N, Li T, Cheng L, Hosseini M, Johnson M and Steedman M (2023) Sources of Hallucination by Large Language Models on Inference Tasks. In: Bouamor H, Pino J and Bali K (eds.) *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, pp. 2758–2774. DOI:10.18653/v1/2023.findings-emnlp.182. URL https://aclanthology.org/2023.findings-emnlp.182.

Paulheim H (2017) Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8(3): 489–508. Publisher: IOS Press.

Ruy FB, Guizzardi G, Falbo RA, Reginato CC and Santos VA (2017) From reference ontologies to ontology patterns and back. *Data & Knowledge Engineering* 109: 41–69. DOI:10.1016/j.datak.2017.03.004. URL https://linkinghub.elsevier.com/retrieve/pii/S0169023X17301076.

Sallam M (2023) ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns. *Healthcare (Switzerland)* 11(6). DOI:10.3390/healthcare11060887. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85151154236&doi=10.3390%2fhealthcare11060887&partnerID=40&md5=254941f18e9afecefc5312194ac61b76. Type: Review.

Savelka J (2023) Unlocking Practical Applications in Legal Domain: Evaluation of GPT for Zero-Shot Semantic Annotation of Legal Texts. In: *19th International Conference on Artificial Intelligence and Law, ICAIL 2023 - Proceedings of the Conference*. pp. 447 – 451. DOI:10.1145/3594536.3595161. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85177829432&doi=10.1145%2f3594536.3595161&partnerID=40&md5=f79b552235236cc2501f5d4099610665. Type: Conference paper.

Shakarian P, Baral C, Simari GI, Xi B and Pokala L (2023) Brief Introduction to Propositional Logic and Predicate Calculus. In: *Neuro Symbolic Reasoning and Learning*. Cham: Springer Nature Switzerland. ISBN 978-3-031-39179-8, pp. 3–14. DOI:10.1007/978-3-031-39179-8_2. URL https://doi.org/10.1007/978-3-031-39179-8_2.

Sharma K, Kumar P and Li Y (2025) OG-RAG: Ontology-grounded retrieval-augmented generation for large language models. In: Christodoulopoulos C, Chakraborty T, Rose C and Peng V (eds.) *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics. ISBN 979-8-89176-332-6, pp. 32962–32981. DOI:10.18653/v1/2025.emnlp-main.1674. URL https://aclanthology.org/2025.emnlp-main.1674/.

Shen W, Wang J and Han J (2015) Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27(2): 443–460. DOI:10.1109/TKDE.2014.2327028.

Vasilecas O (2007) *Databases and Information Systems IV: Selected Papers from the Seventh International Conference DB&IS'2006*. Number v.155 in Frontiers in Artificial Intelligence and Applications Series. Amsterdam: IOS Press. ISBN 978-1-58603-715-4 978-1-60750-226-5.

Wang Q, Mao Z, Wang B and Guo L (2017) Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering* 29(12): 2724–2743. Publisher: IEEE.

Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi EH, Le QV and Zhou D (2022) Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: *Advances in Neural Information Processing Systems*, volume 35. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85163157409&partnerID=40&md5=fbb697b67c926d5bf33b57d921a45bcc. Type: Conference paper.

Yang S, Ma W, Sun P, Zhang M, Ai Q, Liu Y and Cai M (2024) Common Sense Enhanced Knowledge-based Recommendation with Large Language Model. URL http://arxiv.org/abs/2403.18325.

Zhang J, Chen B, Zhang L, Ke X and Ding H (2021) Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open* 2: 14–35. DOI:https://doi.org/10.1016/j.aiopen.2021.03.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000061.

Zuccon G, Koopman B and Shaik R (2023) ChatGPT Hallucinates when Attributing Answers. In: *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. Beijing China: ACM. ISBN 979-8-4007-0408-6, pp. 46–51. DOI:10.1145/3624918.3625329. URL https://dl.acm.org/doi/10.1145/3624918.3625329.