**Solid Freeform Fabrication 2024: Proceedings of the 35th Annual International**
**Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference**
**Reviewed Paper**

# Ontology-based Retrieval Augmented Generation (RAG) for GenAI-supported Additive Manufacturing

Yeun Park[1,2], Paul Witherell[1], Nowrin Akter Surovi[1,3], and Hyunbo Cho[2]

[1]National Institute of Standards and Technology (NIST), USA
[2]Pohang University of Science and Technology (POSTECH), ROK
[3]Singapore University of Technology and Design (SUTD), SGP

## Abstract

Conventional data analytics often fail to capture the intricate context of additive manufacturing (AM) processes, leading to pointed solutions and suboptimal analytics outcomes. The performance of generative artificial intelligence (GenAI) models, such as large language models (LLMs), largely depends on their ability to integrate and contextualize the vast data they are trained on. However, contextualizing is often directly driven by the data consumed, and not necessarily grounded in the fundamental truths. To address this issue, an ontology-based retrieval augmented generation (RAG) approach is proposed to enhance GenAI's capability to generate pertinent prompts and answers. The GenAI recognizes and applies relevant context by leveraging structured ontology, resulting in accurate and insightful interpretations. A use case showcases how the proposed ontology-based RAG framework operates to provide context-aware AM data analytics that promote analytical transparency through fundamental truths when executing AM data analytics.

## 1. Introduction

Additive manufacturing (AM) is an emerging field with significant potential, yet it faces several challenges, from design to process optimization. These challenges include designing the optimal CAD model, selecting appropriate materials, and dealing with defects. Such issues can impact the accuracy and quality of the final product, reducing its strength, lifespan, and performance [1]. Machine learning (ML) models play a vital role in addressing these challenges, but each ML model is typically specialized to solve specific problems. Consequently, a single ML model cannot address all AM-related issues.

In this context, generative artificial intelligence (GenAI) offers a promising solution. GenAI refers to algorithms that generate multimodal content such as text, images, or designs based on learned data patterns, and replicating real data distributions [2], [3]. GenAI models are trained on diverse datasets, capable of handling multimodal data, and can address multiple tasks within AM if adequately trained. Therefore, GenAI can potentially replace multiple specialized models with a single model [4]. These models provide various applications and opportunities in AM, from design to process optimization and beyond, leading to innovative and efficient solutions [5]. Furthermore, using GenAI effectively can help organize and utilize AM-specific data and insights systematically, enhancing the overall data analytics process.

However, the complexity of AM presents significant challenges for the effective use of GenAI, as general GenAI often struggles to capture the intricate context of AM process, data, and materials, which is essential for accurate data analytics. Furthermore, GenAI models are prone to hallucinations, generating irrelevant or unreliable content, leading to misinformation [6]. These hallucinations can result in responses that appear correct but are factually incorrect or even produce fictitious information and fake images. The need for more context awareness and the risk of generating inaccurate information highlights the importance of advanced techniques like prompt engineering [2].

Prompt engineering refers to strategically designing and formulating prompts to align with the model's training and capabilities to improve the accuracy and relevance of the generated answers. By crafting clear, specific, and contextually appropriate questions that match the model's strengths, prompt engineering enhances GenAI model responses. This method optimizes performance without retraining or modifying the model, making it an efficient and cost-effective approach [7].

Building on the concept of prompt engineering, retrieval augmented generation (RAG) is a promising technique for enhancing the generating capabilities of GenAI models by incorporating external knowledge sources as input prompts [6]. By leveraging RAG within prompt engineering, it becomes possible to enhance the accuracy and contextual relevance of GenAI outputs, making GenAI more effective in the complex domain of AM.

This research explores an ontology-based RAG technique that provides effective prompts to enhance GenAI's effectiveness in solving AM data analytics tasks. This approach aims to improve the contextual relevance and accuracy of GenAI-generated responses by integrating AM-specific knowledge into GenAI's processing workflow. By systematically incorporating external knowledge sources through ontology-based RAG, more effective prompts for GenAI will be provided, thereby enhancing its performance in AM data analytics.

The remainder of this paper is organized as follows. Section 2 provides the background of our method, discussing GenAI for data analytics, the application of data analytics in AM, and the role of ontologies for AM. Section 3 explores prompt engineering for GenAI, explaining how well-crafted prompts enhance the performance of GenAI models. Section 4 details the proposed method for applying ontology-based RAG in GenAI-supported AM, describing its components and functionality. Section 5 presents a case study demonstrating the effectiveness of the proposed method. Finally, Section 6 concludes the paper, summarizing our research contributions and outlining potential directions for future work.

## 2. Background

### 2.1 GenAI for Data Analytics

GenAI refers to algorithms capable of generating novel, creative, and realistic content, such as images, audio, video, and 3D models, by replicating real data distributions [8]. GenAI can be used in data analytics in various ways, including data processing and cleaning, synthetic data generation, pattern recognition, prediction, structuring text documents, and analyzing images and videos for anomaly detection [9]. Consequently, GenAI plays a vital role in data analytics.

However, the effective use of GenAI depends largely on guiding the models to the correct solution space. These models require extensive training on large datasets, and the effectiveness of this training can vary based on the selected data. Moreover, even with well-trained models, variations in query formulation can significantly influence the quality of the results. Therefore, before enhancing the capabilities of GenAI models through fine-tuning or retraining, it is crucial to understand what these models are currently capable of by effectively querying them. In this context, the concept of prompt engineering becomes essential.

Prompt engineering is a powerful technique that enhances the quality of responses from GenAI models by crafting the "right questions" effectively. It involves strategically designing and phrasing prompts to align with the model's training and capabilities, thereby improving the accuracy and relevance of the generated answers. By formulating clear, specific, and contextually appropriate queries, prompt engineering helps guide GenAI tools toward producing more precise and valuable responses, addressing the complexities and nuances of the issues at hand. This approach focuses on refining how queries are structured to elicit the most accurate and relevant outputs from the models. By carefully crafting prompts, users can effectively leverage the model's existing knowledge and capabilities, optimizing its performance for specific tasks. This method of enhancing model utility is particularly valuable as it avoids the need for retraining or modifying the model's architecture, making it a cost-effective and efficient way to improve outcomes [10].

### 2.2 Data Analytics in Additive Manufacturing

AM data analytics uses advanced tools and techniques to optimize and control the AM process from design to final product. These analytics are widely applied because they provide actionable insights that improve design, predict material properties, estimate costs, and detect defects, enhancing overall efficiency and quality [11]. However, despite their effectiveness, these data analytics methods often fail to address the complexities of AM projects, where interpreting the intricate relationships and dependencies within AM processes is essential [12].

Understanding the context in which AM data is generated is crucial for more accurate interpretation and analysis. Contextual knowledge, such as machine settings, environmental conditions, and material properties, helps uncover deeper insights and reveals interrelationships within the data, leading to better decision-making. By incorporating contextual information, AM data analytics becomes more precise and reliable, ultimately resulting in more effective and informed outcomes [13].

### 2.3 Ontologies for Additive Manufacturing

An ontology is a formal and explicit description of concepts within a domain, including classes, properties of each class, and restrictions on these properties. It provides a structured framework to organize knowledge, allowing for the definition of relationships between concepts, and serves as a foundation for creating a comprehensive knowledge base by defining individual instances of these classes. Ontologies are used to share a common understanding of the structure of information, enable the reuse of domain knowledge, make domain assumptions explicit, and analyze domain knowledge [14].

Ontologies have been utilized to develop AM knowledge bases that provide context for AM tasks by structuring AM-specific domain knowledge. This structured knowledge makes context explicit, enabling more accurate interpretation and analysis of AM context. Recently, ontologies have been increasingly applied in AM for knowledge representation and management, encompassing areas such as process plans, AM process parameters, product lifecycle, and sensor data [15]. Advanced data analytics tools, including machine learning and knowledge graphs, have been used to develop ontology-based knowledge representations, such as the DfAM ontology [16]. Additionally, ontologies are employed to address specific data analytics tasks in AM, enhancing collaborative knowledge management and providing structured frameworks for more accurate and efficient problem-solving [17]. Furthermore, ontologies can serve as an external knowledge base for RAG models, improving prompts for GenAI and enhancing its performance and reliability in addressing complex AM tasks.

## 3. Prompt Engineering for GenAI

Prompt engineering encompasses a wide range of skills and techniques essential for effectively interacting with and developing GenAI models. It involves more than just designing prompts; it requires configuring various parameters, such as temperature, top-p, and max length, to achieve desirable and reliable responses. This process often involves experimentation to determine the optimal settings for specific use cases. When crafting prompts, several key elements are considered: instructions (specific tasks for the model), context (additional information to guide responses), input data (questions or topics of interest), and output indicators (desired format or type of response). By understanding and utilizing these components, prompts can be tailored to perform various tasks, including text summarization, information extraction, question answering, text classification, conversation, code generation, and reasoning. Learning these concepts is best achieved through practical examples, which demonstrate how well-crafted prompts can effectively address different tasks [18].

However, for more complex and knowledge-intensive tasks, particularly in the field of AM, GenAI models can produce irrelevant responses even with specific parameter settings and well-crafted prompts. These inconsistencies can lead to misinformation, decreased trust in AI systems, and suboptimal decision-making, particularly in complex fields like AM. Therefore, it is important to build a GenAI system that accesses external knowledge sources to complete these tasks. This approach enhances factual consistency, improves the reliability of the generated responses, and helps mitigate the problem of hallucinations [6], [19]. RAG technology, introduced by Meta AI researchers, is a type of GenAI system that enhances prompts for GenAI [2].

RAG enhances GenAI's capabilities by integrating external knowledge sources into the generation process. RAG takes input and retrieves a set of relevant supporting documents from an external knowledge source. These documents are concatenated as context with the original input prompt and fed to the text generator, which produces the final output. This makes RAG adaptive to situations where facts could evolve over time, which is particularly useful given that GenAIs' parametric knowledge is static. RAG allows language models to bypass retraining, enabling access to the latest information for generating reliable outputs via retrieval-based generation.

The general RAG model consists of three main steps: indexing, retrieval, and generation, as shown in Figure 1. In the indexing step, raw data from the external knowledge source is cleaned, segmented into chunks, encoded into vectors, and stored in a vector database. In the retrieval phase, a user query is transformed into a vector and matched with the most similar chunks, which are then used as context in the generation phase to generate a response based on the query and retrieved documents [15]. This approach is particularly useful for addressing knowledge-intensive tasks. RAG can be fine-tuned, and its internal knowledge can be efficiently modified without the need to retrain the entire model [20].

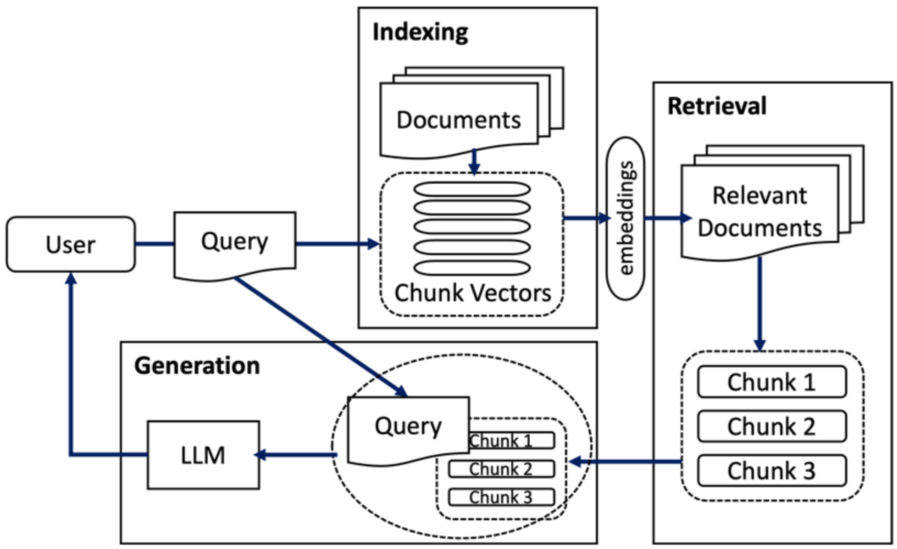

Figure 1. General RAG Model

RAG allows GenAI models to access and utilize up-to-date, domain-specific information, reducing the likelihood of generating hallucinations or incorrect data [2]. In the context of AM, where precision and reliability are important, RAG can significantly enhance the quality of data analytics and decision-making processes. Applying RAG with GenAI ultimately leads to more consistent, reliable, and trustworthy GenAI responses, improving their overall effectiveness in specialized domains like AM.

Ontology-based RAG is one of the RAG techniques that enhances GenAI's generating capability by integrating structured domain knowledge into the retrieval and generation process. Incorporating ontology into RAG enhances the accuracy of retrieved information, ensures consistency in responses, and enables complex queries to be handled by structuring and managing domain knowledge effectively, allowing GenAI to access comprehensive and precise information. Unlike general RAG methods that typically focus solely on text-based entity retrieval, this approach maintains a keen awareness of graph topology, which is essential for generating contextually and factually coherent responses [21]. Therefore, ontology-based RAG enhances the precision and relevance of the generated responses, making it particularly suitable for complex fields such as AM.

## 4. Ontology-based RAG for GenAI-supported Additive Manufacturing

The proposed method uses the AM data analytics (AMDA) ontology as an external knowledge source to enhance AM data analytics by providing structured and contextually rich information into the GenAI model [12]. Figure 2 shows this method has four main functions: Ontology-to-Graph Converter, Entity Extractor, Graph Retriever, and Prompt Generator. The Ontology-to-Graph Converter transforms the AMDA ontology into a comprehensive knowledge graph, structuring the context of AMDA-specific information, as a graph enables efficient querying and relationship mapping for the RAG model. The Entity Extractor processes user queries to identify relevant AM concepts within the query. The Graph Retriever searches the knowledge graph to find and extract the most pertinent subgraphs, providing the necessary context for the query. Finally, the Prompt Generator uses these subgraphs and the query to generate detailed and contextually rich prompts for the GenAI model, enabling it to generate contextually appropriate responses for AM data analytics.

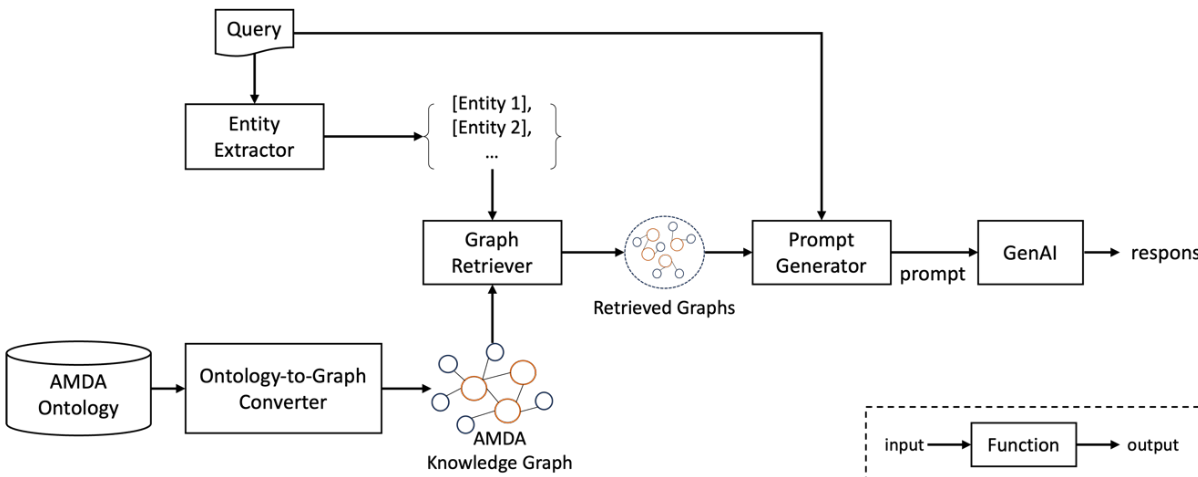


Figure 2. The Framework of Ontology-based RAG for GenAI-supported AM

### 4.1 Ontology Design for RAG

To integrate an ontology into the RAG framework, the ontology should first be transformed into a graph structure, requiring consideration of the differences between ontology and graph representations during the design phase. As explained in Section 2.3, an ontology consists of classes, properties, and restrictions, which provide a structured representation of concepts and their interrelationships. However, a graph is primarily composed of nodes and edges, where nodes represent concepts and edges represent relationships between these nodes. Each node and edge have defined attributes that provide additional context and detail. Therefore, when converting an ontology to a graph, it is crucial to ensure that the necessary context is accurately reflected in the graph representation. This requires designing the ontology with a focus on graph conversion, ensuring that all relevant information is preserved and effectively utilized in the graph structure.

In an ontology, relationships are captured through properties that require traversing multiple connections of classes and instances to gather the full context of an entity. In contrast, graphs benefit from having nodes with direct edges, which reduces the complexity of data retrieval of an entity. Therefore, when converting an ontology to a graph for RAG applications, it is crucial to design the structure to maximize contextual connections around key entities. This ensures that the context within the ontology is accurately transformed and effectively utilized in the graph, enhancing the RAG's ability to deliver accurate and contextually relevant information to the generator.

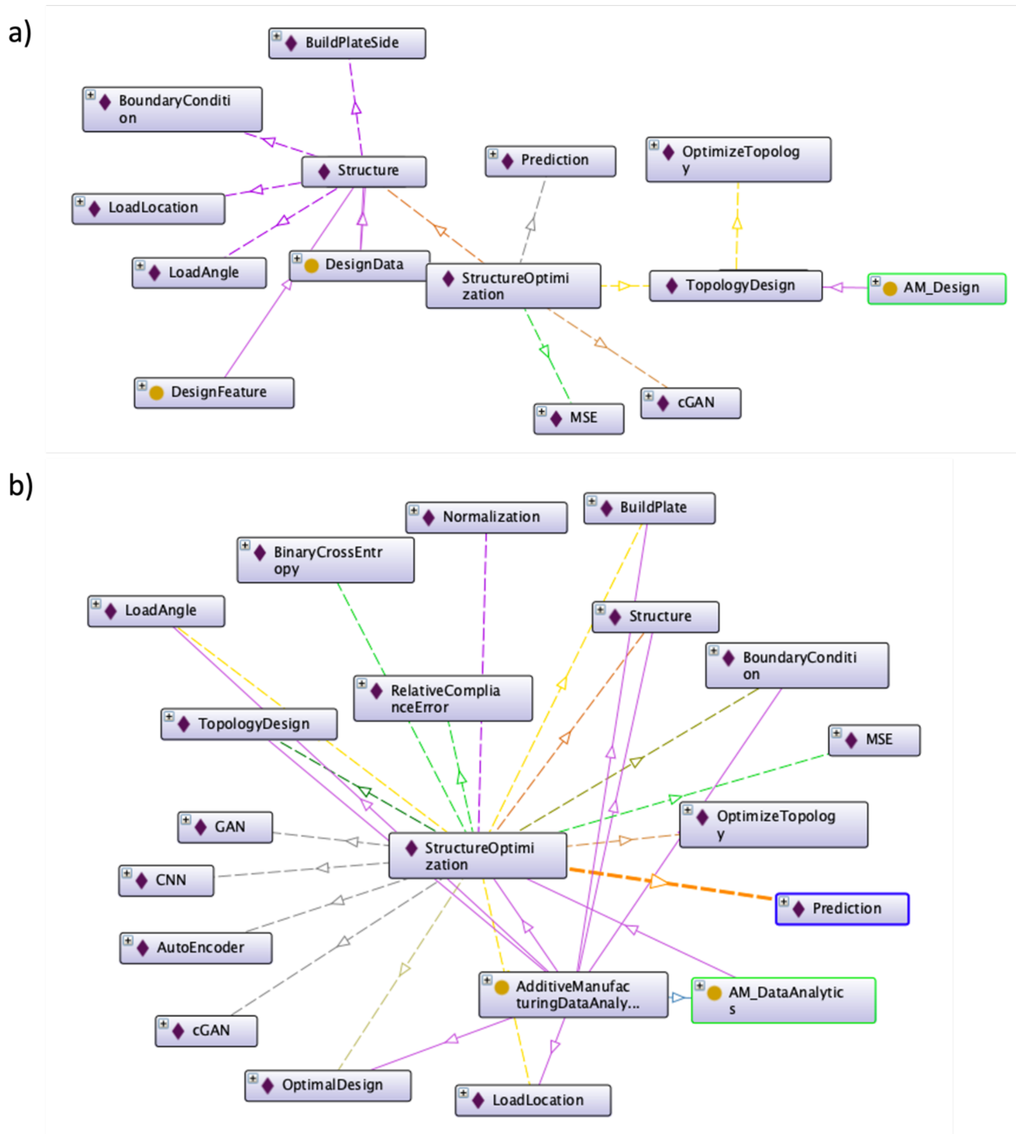


Figure 3. a) Structure of General Hierarchical Ontology,
b) Structure of Ontology for RAG Application

RAG requires searching for specific entities, making it advantageous for each entity to be highly connected to its relevant context. As previously discussed, having more direct edges in a graph ensures that all relevant information is easily accessible. This high level of connectivity allows for efficient retrieval of comprehensive context about an entity, facilitating more informed generations. This differs from an ontology, where relationships might be more abstract and hierarchical, potentially requiring additional steps to interpret fully and retrieve all relevant information. As shown in Figure 3, the differences between the two ontology design structures are clear. Figure 3 a) illustrates the ontology with a hierarchical structure, where relationships are abstract and require traversing multiple connections to gather the full context of an entity, making it less efficient for quick data retrieval. In contrast, Figure 3 b) depicts the ontology structure optimized for RAG application, with entities like "Structure Optimization" having direct connections to related nodes such as "Load Angle," "Load Location," "Build Plate Side," and "Structure". This ensures that all pertinent data is quickly accessible, enabling the framework to provide better responses.

## 4.2 Ontology-to-Graph Converter

The Ontology-to-Graph Converter transforms the structured knowledge from the AMDA ontology into a comprehensive knowledge graph. It begins by extracting classes and individuals from the ontology and representing them as nodes. Each class is identified by its name and is added to the graph only once to avoid duplication. Similarly, individuals are extracted and associated with their respective class types, ensuring the graph captures the complete classification hierarchy. Relationships between classes, such as subclass hierarchies, and object properties between individuals are converted into edges. This representation allows for efficient querying and retrieval of domain-specific information. The resulting knowledge graph serves as a foundation for subsequent steps in the framework, enabling precise and contextually enriched data analytics.

As shown in Figure 4, the knowledge graph contains nodes representing various entities such as "StructureOptimization", "LoadLocation", "MSE" (Mean Squared Error), and "Normalization". Each node is detailed with its properties and relationships to other nodes, illustrating the interconnected nature of the information. For example, the node for "MSE" has properties such as "Data Analytics," "Data Science Context," and "Model Evaluation," explaining the context of "MSE." It also connects to multiple relevant entities, such as "Structure Optimization" and "cGAN". This highly connected structure allows for comprehensive context retrieval, enhancing the ability of the framework to generate accurate and relevant responses for AM data analytics tasks.

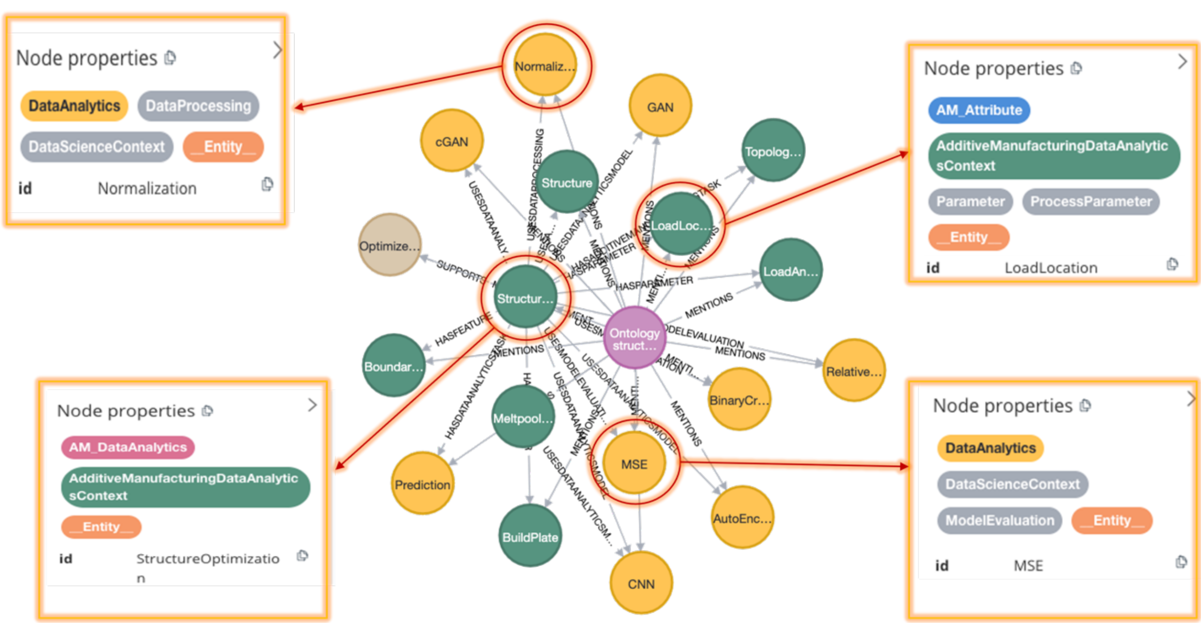

Figure 4. Knowledge Graph (Partial)

### 4.3 Entity Extractor

The Entity Extractor processes user queries to identify and extract relevant AM concepts and entities. Utilizing GenAI, the extractor recognizes key terms related to AM, such as "material properties", "process parameters", and "machine settings", and accurately captures them from the query text. Once the entities are identified, they can be utilized as an input for the Graph Retriever.

A prompt template is designed to guide this extraction process, ensuring consistency and accuracy in the extraction. This template sets the context for GenAI, instructing it to extract AM-specific entities from the given query text. The prompt template includes both system and human messages to facilitate this process. The system message directs GenAI to extract "additive manufacturing", "data analytics", and "additive manufacturing data analytics" entities from the text. The human message provides the format of the query to extract informatiom. The extractor processes the query text, identifies the relevant entities, and formats the output into a list of extracted entities. This approach allows for efficient processing of user queries, ensuring that the extracted entities are both accurate and contextually relevant, thereby enhancing the overall performance of the framework.

### 4.4 Graph Retriever

The Graph Retriever searches the knowledge graph to find and extract relevant subgraphs to the user query. It begins by processing the input query to create a full-text search query, which involves splitting the query into individual words and allowing for minor misspellings. This ensures the search can tolerate some errors in the input, enhancing the robustness of the retrieval process. After extracting entities from the user query, the retriever generates a full-text search for each entity on the knowledge graph. This involves searching for nodes in the graph that match the entities and identifying nodes and their scores based on relevance. The retriever then examines the relationships of these nodes, identifying connected nodes through both incoming and outgoing relationships, thereby mapping out the neighborhood of each identified entity.

By identifying and extracting the subgraphs that best match the query, the Graph Retriever provides the necessary context for generating accurate responses. This approach ensures that the retrieved information is both relevant and comprehensive. By efficiently mapping the query entities to the knowledge graph, the Graph Retriever bridges the gap between user queries and the structured knowledge graph. This process enhances the overall performance of the framework by ensuring that the retrieved information is contextually relevant and accurately reflects the relationships within the knowledge graph.

### 4.5 Prompt Generator

The Prompt Generator combines the extracted subgraphs and the user query to create detailed and contextually rich prompts for the GenAI. A predefined template is used to integrate the extracted subgraphs and the user query into a structured prompt. This structured prompt is then presented to the GenAI, which uses the detailed context to generate a response. This prompt includes both the user's query and the context provided by the knowledge graph, ensuring that the GenAI model has all the necessary information to generate accurate responses. By incorporating structured knowledge into the prompts, the Prompt Generator enhances the GenAI's ability to provide precise and contextually appropriate answers. This step is essential for leveraging the full potential of the ontology-based RAG framework.

### 4.6 Implementation of the proposed method

Implementing the proposed method involves integrating the Ontology-to-Graph Converter, Entity Extractor, Graph Retriever, and Prompt Generator into a cohesive system. First, the AMDA ontology is developed and maintained using Protégé, a widely-used ontology editor [22]. The ontology is then converted into a comprehensive knowledge graph using Neo4j, which provides robust graph database capabilities for managing and querying complex graph data [23]. User queries are processed by the entity extractor, which is implemented using LangChain. This component is designed to identify and extract relevant concepts from the user input, ensuring that the key terms and phrases are accurately captured. These identified concepts are then utilized by the Graph Retriever, also implemented using LangChain, to search the Neo4j knowledge graph for the most relevant subgraphs [24]. The extracted subgraphs and the original user query are then input into the Prompt Generator. This component creates detailed and contextually rich prompts tailored for the GenAI model, powered by OpenAI's GPT-4-turbo [25]. Finally, the GenAI model uses these prompts to generate accurate and contextually appropriate responses, enhancing AM data analytics.

## 5. Case Study

In the case study, the proposed ontology-based RAG framework was implemented and demonstrated through step-by-step operation using the "structure optimization" use case to create the AMDA ontology [13], [26]. This framework was implemented, and its responses were compared to a general GenAI model. Specifically, the performance of handling a query related to AM data analytics was assessed and compared to that of a general GenAI model. The general GenAI system uses OpenAI's GPT-4-turbo model without additional contextual information, while our proposed framework leverages structured domain knowledge from the AMDA ontology to provide more accurate and relevant responses [25].

Both systems were implemented to conduct the case study, and a query relevant to specific AM data analytics tasks was formulated. The query used was "How to implement a data analytics model for structure optimization?". The Entity Extractor then identified key concepts from the query: 'data analytics' and 'structure optimization'. Using these concepts, the Graph Retriever searched the knowledge graph and extracted related subgraphs for each entity.

As shown in Figure 5, the extracted subgraphs provide a detailed map of relevant entities and their relationships. The subgraphs include nodes related to data analytics, structure optimization, and their properties. The format of expression used is *(subject node) - (Relations)->(object node)* or *(object node) <- (Relations) <- (subject node)* or *(node) - (node property)*. For example, the subgraph includes relationships such as *Data Analytics – SUBCLASS OF -> Data Science Context* and *Structure Optimization – USES DATA ANALYTICS MODEL -> GAN*. Node properties such as Data Analytics - Class and MSE – Error Metrics are also detailed.

The extracted subgraphs and the user query were then input into the Prompt Generator. The GenAI model used these prompts to generate accurate and contextually appropriate responses. For the query "How to implement a data analytics model for structure optimization?", the GenAI suggested data analytics models, model evaluation methods, and specific parameters and features that are relevant to the structure optimization task. The case study demonstrated how the system effectively contextualizes and retrieves pertinent information, leading to more accurate and relevant responses.

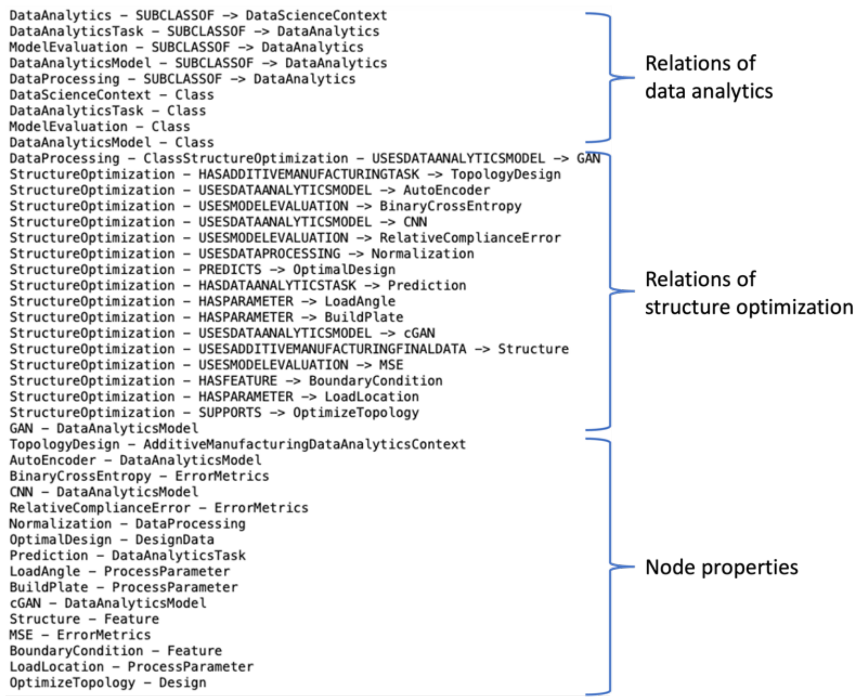

*Figure 5.* Retrieved Subgraphs from Graph Retriever

Additionally, the responses to the query from the general GenAI model and the proposed framework were compared. As shown in Table 1, The general GPT-4-turbo model provided a broad and less detailed response. It suggested general steps like gathering data, preprocessing, and selecting models without specific recommendations. The response did not reference any particular data analytics models, evaluation metrics, or parameters and lacked context-specific advice tailored to AM data analytics tasks.

In contrast, our proposed framework provided a more detailed and contextually rich response. It suggested specific data analytics models such as GANs, AutoEncoders, CNNs, and cGANs, which are known for handling complex spatial data and generating optimized designs. The framework recommended data processing techniques like normalization to prepare the dataset for analysis, ensuring the input data is scaled and formatted correctly for model training. It also provided model evaluation metrics such as binary cross-entropy, relative compliance error, and mean squared error for assessing model performance. Moreover, our framework included specific parameters and features relevant to the structure optimization task, such as load angle, build plate, load location, and boundary conditions. This detailed and context-aware answer demonstrated the effectiveness of the ontology-based RAG framework in enhancing AM data analytics by leveraging structured domain knowledge, leading to more accurate and relevant responses.

Table 1. Responses from the General GenAI and Our Proposed Framework

| Query | How to implement data analytics model for Structure optimization? | |
|---|---|---|
| **Model** | **GPT4** | **Ontology-based RAG + GPT4 (Proposed Framework)** |
| **Response** | To implement a data analytics model for structure optimization in additive manufacturing, start by collecting comprehensive data from the manufacturing process, including material properties, machine parameters, and geometric specifications of the parts. Next, use machine learning algorithms such as regression analysis, neural networks, or genetic algorithms to analyze the relationships between these parameters and the resulting structural performance. Develop a predictive model that can simulate different manufacturing scenarios and predict their impact on the structure's performance. Integrate this model into a design optimization tool that iteratively adjusts design parameters to optimize structural integrity, weight, and material usage. Finally, validate the model by comparing its predictions with experimental results and refine it based on feedback to improve accuracy and reliability. | To implement a data analytics model for structure optimization in additive manufacturing, start by selecting appropriate data analytics models such as GANs, AutoEncoders, CNNs, and cGANs, which are known to be effective for handling complex spatial data and generating optimized designs. Next, integrate data processing techniques like normalization to prepare your dataset for analysis, ensuring that the input data is scaled and formatted correctly for model training. Utilize model evaluation metrics such as Binary Cross Entropy, Relative Compliance Error, and Mean Squared Error to assess the performance of your models, allowing for iterative improvements based on quantitative feedback. Incorporate specific parameters and features relevant to the structure optimization task, such as load angle, build plate, load location, and boundary conditions, to tailor the model to the specific requirements of the topology design. Finally, continuously refine the model by leveraging prediction tasks to forecast optimal designs and using the feedback to enhance model accuracy and design efficiency. |

## 6. Conclusion

This research demonstrates the effectiveness of integrating ontology-based RAG with GenAI models to enhance data analytics in AM. By leveraging structured domain knowledge from the AMDA ontology, our proposed framework improves the contextual relevance and accuracy of generated responses compared to the general GenAI model. The case study highlighted the framework's ability to provide detailed, context-aware answers, suggesting domain-specific responses for an AM data analytics task. The proposed framework addresses the limitations of general GenAI systems by incorporating comprehensive, domain-specific information, thereby enhancing the utility of GenAI in complex and knowledge-intensive domains like AM.

Despite the promising results, this research has several limitations. The current ontology may not comprehensively cover all aspects of AM processes, potentially limiting the framework's applicability in diverse AM scenarios. Additionally, the performance of the GenAI model depends on the accuracy and completeness of the AMDA ontology, as any gaps or inaccuracies could negatively impact its effectiveness.

Future work will focus on refining the ontology, expanding its coverage to include more aspects of AM processes, and integrating additional external knowledge sources to further improve the framework's capabilities. Additionally, comprehensive evaluations of various RAG technique will be conducted to better understand its strengths and limitations across different contexts and tasks. Furthermore, handling complex, multi-hop questions within the framework will also be addressed to show its effectiveness in managing intricate queries.

## Acknowledgements

This work was supported by the Additive Manufacturing Program at the National Institute of Standards and Technology (NIST), U.S. Department of Commerce. Such commercial systems identified in this article do not imply recommendation or endorsement by NIST. Moreover, the contents of this article do not reflect the views of NIST or any other U.S. government.

## Reference

[1] Surovi, N. A., & Soh, G. S. (2023). Acoustic feature based geometric defect identification in wire arc additive manufacturing. Virtual and Physical Prototyping, 18(1), e2210553.
[2] Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K., & Chen, L. (2023). Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. Journal of Information Technology Case and Application Research, 25(3), 277-304.
[3] Badini, S., Regondi, S., Frontoni, E., & Pugliese, R. (2023). Assessing the capabilities of ChatGPT to improve additive manufacturing troubleshooting. Advanced Industrial and Engineering Polymer Research, 6(3), 278-287.
[4] Surovi, N. A., Witherell, P., Mathew, V. S., & Kumara, S. (2024, August). Current State and Benchmarking of Generative Artificial Intelligence for Additive Manufacturing. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference . American Society of Mechanical Engineers (Accepted).
[5] Bendoly, E., Chandrasekaran, A., Lima, M. D. R. F., Handfield, R., Khajavi, S. H., & Roscoe, S. (2023). The role of generative design and additive manufacturing capabilities in developing human–AI symbiosis: Evidence from multiple case studies. Decision Sciences.
[6] Li, J., Yuan, Y., & Zhang, Z. (2024). Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. arXiv preprint arXiv:2403.10446.
[7] Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023, June). Prompt engineering in large language models. In International conference on data intelligence and cognitive informatics (pp. 387-402). Singapore: Springer Nature Singapore.
[8] Sakirin, T., & Kusuma, S. (2023). A survey of generative artificial intelligence techniques. Babylonian Journal of Artificial Intelligence, 2023, 10-14.
[9] Yan, L., Martinez-Maldonado, R., & Gasevic, D. (2024, March). Generative artificial intelligence in learning analytics: Contextualising opportunities and challenges through the learning analytics cycle. In Proceedings of the 14th Learning Analytics and Knowledge Conference (pp. 101-111).
[10] Surovi, N. A., & ., Witherell, P. (2024). Generative Artificial Intelligence (GenAI) Prompt Engineering for Additive Manufacturing (AM). (Submitted with Acceptance)
[11] Razvi, S. S., Feng, S., Narayanan, A., Lee, Y. T. T., & Witherell, P. (2019, August). A review of machine learning applications in additive manufacturing. In International design engineering technical conferences and computers and information in engineering conference (Vol. 59179, p. V001T02A040). American Society of Mechanical Engineers.
[12] Park, Y., Witherell, P., Jones, A., & Cho, H. (2023, August). Knowledge Management for Data Analytics in Additive Manufacturing. In International Design Engineering Technical

Conferences and Computers and Information in Engineering Conference (Vol. 87295, p. V002T02A053). American Society of Mechanical Engineers.
[13] Park, Y., Witherell, P., & Cho, H. (2024, August). Ontology-based Context-aware Data Analytics in Additive Manufacturing. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers.
[14] Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.
[15] Ali, M. M., Rai, R., Otte, J. N., & Smith, B. (2019). A product life cycle ontology for additive manufacturing. Computers in Industry, 105, 191-203.
[16] Kim, S., Rosen, D. W., Witherell, P., & Ko, H. (2019). A design for additive manufacturing ontology to support manufacturability analysis. Journal of Computing and Information Science in Engineering, 19(4), 041014.
[17] Park, H., Ko, H., Lee, Y. T. T., Feng, S., Witherell, P., & Cho, H. (2023). Collaborative knowledge management to identify data analytics opportunities in additive manufacturing. Journal of Intelligent Manufacturing, 1-24.
[18] Prompt Engineering Guide. https://www.promptingguide.ai/
[19] Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K., & Chen, L. (2023). Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. Journal of Information Technology Case and Application Research, 25(3), 277-304.
[20] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.
[21] Hu, Y., Lei, Z., Zhang, Z., Pan, B., Ling, C., & Zhao, L. (2024). GRAG: Graph Retrieval-Augmented Generation. arXiv preprint arXiv:2405.16506.
[22] Musen, M. A. (2015). The protégé project: a look back and a look forward. AI matters, 1(4), 4-12.
[23] Technology, Inc. (2015). Neo4j, the World's Leading Graph Database. Neo4j Graph Database.
[24] Langchain. https://www.langchain.com/
[25] OpenAI. (2024). ChatGPT [Large language model].
[26] Hertlein, N., Buskohl, P. R., Gillman, A., Vemaganti, K., & Anand, S. (2021). Generative adversarial network for early-stage design flexibility in topology optimization for additive manufacturing. Journal of Manufacturing Systems, 59, 675-685.