

Article

Quantifying Relational Exploration in Cultural Heritage Knowledge Graphs with LLMs: A Neuro-Symbolic Approach for Enhanced Knowledge Discovery

Mohammed Maree 

Faculty of Information Technology, Arab American University, Jenin 00970, Palestine; mohammed.maree@aaup.edu; Tel.: +970-4-2418888 (ext. 1160)

Abstract: This paper introduces a neuro-symbolic approach for relational exploration in cultural heritage knowledge graphs, exploiting Large Language Models (LLMs) for explanation generation and a mathematically grounded model to quantify the interestingness of relationships. We demonstrate the importance of the proposed interestingness measure through a quantitative analysis, highlighting its significant impact on system performance, particularly in terms of precision, recall, and F1-score. Utilizing the Wikidata Cultural Heritage Linked Open Data (WCH-LOD) dataset, our approach achieves a precision of 0.70, recall of 0.68, and an F1-score of 0.69, outperforming both graph-based (precision: 0.28, recall: 0.25, F1-score: 0.26) and knowledge-based (precision: 0.45, recall: 0.42, F1-score: 0.43) baselines. Furthermore, the proposed LLM-powered explanations exhibit better quality, as evidenced by higher BLEU (0.52), ROUGE-L (0.58), and METEOR (0.63) scores compared to baseline approaches. We further demonstrate a strong correlation (0.65) between the interestingness measure and the quality of generated explanations, validating its ability to guide the system towards more relevant discoveries. This system offers more effective exploration by achieving more diverse and human-interpretable relationship explanations compared to purely knowledge-based and graph-based methods, contributing to the knowledge-based systems field by providing a personalized and adaptable relational exploration framework.



Academic Editors: Xujuan Zhou, Yuefeng Li, Raj Gururajan, Ji Zhang and Revathi Venkataraman

Received: 17 February 2025

Revised: 1 April 2025

Accepted: 8 April 2025

Published: 10 April 2025

Citation: Maree, M. Quantifying Relational Exploration in Cultural Heritage Knowledge Graphs with LLMs: A Neuro-Symbolic Approach for Enhanced Knowledge Discovery. *Data* 2025, 10, 52. <https://doi.org/10.3390/data10040052>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: knowledge graphs; large language models (LLMs); explainable AI (XAI); cultural heritage; neuro-symbolic AI; interestingness score; contextual relevance; personalized explanation

1. Introduction

The digital age has ushered in an unprecedented era of data proliferation, significantly impacting how we access, preserve, and interact with cultural heritage (CH). The digitization of cultural heritage artifacts, historical records, and intangible cultural expressions has resulted in massive repositories of interconnected information, often formalized as knowledge graphs (KGs) [1,2]. These knowledge graphs, while valuable, present a significant challenge: how do we effectively navigate, explore, and extract meaningful insights from this intricate web of interconnected data? Simple keyword searches often prove insufficient, failing to capture the nuanced relationships and hidden narratives woven within these datasets [3]. For example, a simple query for “paintings from the 18th century” might not reveal the connections between artists, movements, and places that shaped artistic production during that period. Thus, the field of relational search—the discovery of connections and relationships between entities within these knowledge graphs—becomes paramount. This goes beyond mere data retrieval, enabling us to uncover the complex

patterns, hidden connections, and narratives embedded in our cultural heritage that would otherwise remain obscured. Relational search can reveal relationships such as the influence of different cultures on the work of an artist, the political and historical context of certain events, and the evolution of artistic styles across time.

Traditional approaches to relational search in cultural heritage KGs have primarily relied on two main paradigms: graph-based methods and knowledge-based approaches. Graph-based methods, such as Breadth-First Search (BFS) or Depth-First Search (DFS), while able to explore the KG structure systematically [4,5], treat the graph as a simple network of nodes and edges, ignoring the rich semantic meaning encoded in the data. These methods may retrieve a large number of paths between entities, but many of them are semantically trivial or irrelevant to a user's query. They also lack the adaptability needed to address the needs of users with diverse backgrounds and interests. For example, BFS might present simple connections like "artist A was born in place B" but fail to highlight more insightful connections such as "artist A's work was influenced by the art of place C", or "artist A's work was part of the art movement of place D". These methods lack the ability to provide a deeper understanding of the relations between entities. On the other hand, knowledge-based approaches [6,7] leverage ontologies, predefined rules, and structured vocabularies (e.g., using CIDOC CRM or SPARQL-based methods) to perform a relational search. Although providing more accurate results by incorporating domain knowledge, they are limited by the rigid scope of the predefined rules and templates. They require significant manual engineering and are often brittle and difficult to adapt to evolving needs, diverse user requirements, and emerging new insights. These methods, while improving on the semantic understanding, fail to provide the flexibility and adaptability needed to uncover the diverse and nuanced relationships found in complex cultural heritage data, as they often rely on simple text templates to present the relationships. They also fail to take into account the user needs and expertise. For instance, a user with advanced knowledge of a specific artist might be interested in very specific types of relationships, while other users may only be interested in more general types of relationships. Moreover, both traditional graph-based and knowledge-based methods fall short in one critical aspect: personalized and human-understandable explanations. They often produce results in forms that are difficult to interpret by an end-user, such as a list of entities or graph paths. The lack of context and meaningful explanations makes it difficult for users to grasp the full implications of the discovered relationships. Existing systems also typically ignore the diverse expertise of end-users, failing to deliver results that are truly relevant to their individual needs and interests. For example, explanations based on predefined rules or simple path traversals fail to convey the richness and complexity of the relationship and fail to adapt to the user needs.

Recognizing these limitations, this paper introduces a novel approach combining the capabilities of Large Language Models (LLMs) with a mathematical model for quantifying the "interestingness" of discovered relationships. The motivation behind this research stems from the need to move beyond systems that simply present data and create a system that actively guides the user to discover relevant connections and explains them in a personalized and human-understandable way. We aim to create a system that is able to learn complex relationship patterns, adapt to the specific needs of end-users, and generate high-quality and human-understandable explanations. The key insight of our work lies in the observation that relational search requires not only the discovery of relationships but also the generation of human-interpretable explanations that are contextually relevant and tailored to the user. Unlike approaches that focus solely on object classification or simple link prediction, we aimed to create a comprehensive system for the generation of high-quality explanations

that were based on the relationships between different entities. Our specific objectives were to:

- * Generate contextually relevant and diverse explanations: LLMs are trained on massive text corpora, giving them a much richer understanding of language and context than traditional NLP models. They can grasp the subtle relationships between concepts and generate explanations that are more nuanced and insightful. As such, we leveraged the power of LLMs to produce more diverse and flexible explanations that go beyond predefined templates. These explanations were able to adapt to the specific semantic context of the relationships, enhancing the user's comprehension of the discovered information. We aimed at creating explanations that go beyond simple information and provide deep insights about the discovered relationships.
- * Personalize explanations by considering user context: LLMs are designed to provide explanations that are much easier for users to understand compared to graph paths or complex SPARQL queries. We achieved this by prompting the LLM with structural text that contains connection information as well as context and letting the LLM create a response based on the structural text. In this context, we aimed at creating a system that provides personalized explanations by considering the user's current context, preferences, knowledge, and past interactions, ensuring that the presented explanations are both relevant and engaging. This is achieved by allowing the user to define specific parameters or interests, as well as implicitly by taking into account past interactions and expertise.
- * Quantify relationship interestingness using a mathematical model: We introduced a novel mathematical model to assess the interestingness of relationships. This model, which combines measures of semantic relatedness within the graph and contextual relevance to a specific user, enabled our system to effectively guide the exploration process, revealing non-trivial and valuable connections. This provided an objective and data-driven approach to guide the exploration process, moving beyond simple relevance and enabling the discovery of non-obvious connections.

The rest of this article is structured as follows. Section 2 reviews existing relational search methods, highlighting their limitations. It also introduces our methodology, detailing connection discovery, the mathematical model for scoring interestingness, and our use of LLMs for explanation generation. Section 3 describes the experimental setup, including dataset information, implementation specifics, and evaluation procedures. In this section, we also present and discuss the quantitative performance metrics. Finally, Section 4 summarizes our conclusions and outlines avenues for future research.

2. Materials and Methods

The challenge of uncovering meaningful connections within knowledge graphs (KGs) has spurred a diverse range of research efforts [2,5,8–24]. In this section, we explore and discuss existing techniques, grouping them into relevant categories with an emphasis on their strengths and limitations. We have organized these techniques into knowledge-agnostic graph traversal methods, knowledge-based approaches, explainable AI in knowledge graphs, and the emerging use of LLMs in knowledge graph reasoning.

2.1. Knowledge-Agnostic Graph Traversal

These methods operate on the structural properties of the KG, treating it as a network of interconnected nodes and edges. They aim to find relationships between entities without relying on the semantic meaning of the data itself. This makes them generalizable but also prone to generating a lot of irrelevant results [8]. In addition, these approaches are based on simple graph traversal methods. For example, using fundamental algorithms

such as the Breadth-First Search (BFS) and Depth-First Search (DFS), relationships between nodes are systematically explored in KGs [9]. BFS starts from a source node and visits all its neighbors before moving to the next level, while DFS explores as far as possible along each branch. Although these approaches can find all reachable nodes or paths between nodes, they are computationally expensive in large KGs and frequently generate numerous paths that are semantically trivial. They often produce a combinatorial explosion of results that are uninteresting or obvious to a user, requiring manual post-processing and filtering [10]. Other algorithms, like Dijkstra's and A*, find the shortest path between two entities based on edge weights or hop counts. Methods like WiSP [11] can be used to find the most direct paths, but still do not capture semantic relatedness or relevance to the user's context. Although these algorithms can find the most direct connection between entities, that path may not always be the most relevant, surprising, or informative for relational exploration, and might be too simplistic to capture the nuances of the relationships [12].

On the other hand, Random Walk and Path Ranking techniques simulate a random walk across the graph to compute node importance and explore connectivity. Methods like Personalized PageRank use random walks to identify relevant results for specific nodes; however, while these methods are scalable and more efficient than other graph traversal methods, they do not typically capture the rich semantics of relationships and user context [13]. They also can produce many irrelevant and uninterpretable paths. Accordingly, we would like to point out that, while efficient for exploring the structure of KGs, these knowledge-agnostic methods fall short in addressing the core problem of relational search. They are not guided by any form of domain understanding, and therefore, often generate a plethora of irrelevant results and provide limited explanations that lack context. The paths are not always human understandable, and do not provide deep explanations of the relationship discovered.

2.2. Knowledge-Based Approaches

These techniques leverage domain-specific knowledge, often encoded in ontologies, rules, and schemas, to guide the relational exploration, but at the expense of flexibility and manual engineering [2,14]. These techniques aim to mitigate some of the limitations of purely graph-based approaches by incorporating more structure and semantics. For instance, ontology-based search methods use formal ontologies (e.g., CIDOC CRM) to reason about relationships between entities [2]. While ontologies provide a structured vocabulary for representing knowledge, they are mostly designed for structured data retrieval and do not provide the flexibility or ability to generate natural language explanations that are tailored to a user. They are often more suitable for querying structured data instead of providing explanations. On the other hand, SPARQL-based approaches use the SPARQL query language to specify precise patterns of interest [15]. SPARQL offers a powerful way to extract complex data from RDF graphs. However, such approaches often rely on manually created queries and predefined templates for explanations, which are brittle and hard to adapt to varying user needs or different contexts. In a similar line of research, rule-based methods use predefined rules to infer and discover new relations [5]. Although they can provide some level of domain-specific reasoning, they are generally limited by the scope of the rules and their lack of adaptability. Moreover, these rules need to be specified by domain experts which increases the workload in creating, maintaining, and adapting them for different scenarios. These approaches address precision to some extent but are limited by the scope of predefined rules and templates, as well as their lack of personalization. As noted in our introduction, they fail to capture the nuances of relationships and are not adaptable to varying user contexts [16,17]. Accordingly, these approaches lack the ability

to provide flexible explanations for a varied number of users in real-world application contexts and settings.

2.3. Explainable AI in Knowledge Graphs

Explainable AI (XAI) in the context of KGs aims to provide human-understandable explanations for the relationships discovered in the graphs. However, this area is still in early stages and faces some limitations. For instance, in path-based explanation methods, like RelFinder [18], connections are visualized as graph paths or subgraphs connecting two or more entities. However, these paths and subgraphs might be hard to interpret and understand, especially for users who are not domain experts, or have no technical background. These methods also do not provide natural language explanations [19]. Other subgraph-based explanation methods extract a portion of the knowledge graph to create an explanation that provides more context than simple paths [20]. However, subgraphs still require some effort for interpretation by end users, as it is not presented in natural language. In addition, as reported in [21], such methods require substantial computational resources, leading to extensive utilization of approximation techniques and other optimization approaches. Other Graph Neural Networks (GNNs) have also been applied for reasoning and link prediction [22,23]. While GNNs can learn complex patterns and produce good results on link prediction, they are also very complex to interpret, as they often operate as black boxes and are not suitable for generating natural language explanations to the end-user [24]. As reported in a recent review in [25], there are four open problems, including robustness, interpretability, pretraining, and complex structure modeling, that still form a major challenge for GNNs and hinder their exploitation in real-world and practical application domains, such as the CH domain. Therefore, while these methods try to address the lack of transparency of AI systems, they still struggle to create rich, human-readable, and context-aware explanations that go beyond just graph paths or subgraphs. They often lack the ability to tailor explanations to the context or knowledge level of the user and require a level of technical expertise.

2.4. Large Language Models in Knowledge Graph Reasoning

The use of Large Language Models (LLMs) in KG applications is an emerging field. Although LLMs are showing great potential in many other areas, their integration with relational search in KGs is still in early stages. For instance, LLMs can answer questions based on the information available in KGs. They can extract relevant entities, use reasoning and language understanding, and provide answers to natural language questions. Moreover, they can be trained on KG embeddings for performing link prediction, by learning representations of nodes and relationships. However, these approaches often focus on the accuracy and efficiency of link prediction but do not focus on generating high-quality explanations. On the other hand, recent research attempts have explored methods for fusing neural networks with symbolic knowledge representations for explainable AI. These approaches, including the work of Díaz-Rodríguez et al. [26] and Arrieta et al. [27], combine a neural network for image classification with a knowledge graph to guide the detection of object-parts.

These approaches use a knowledge graph and a neural network in a fused way to train the neural network, where the knowledge graph is used as a form of regularization. However, they do not make use of an LLM for generating natural language explanations, nor do they use a user context for tailoring explanations. They also do not focus on relational exploration, but rather on improving the performance and explainability of object classification through the use of a knowledge graph. Therefore, our goal is to go beyond simply retrieving data from the knowledge graph. Instead, we aim to generate

human-understandable explanations of connections and guide the search towards more interesting relationships, facilitating deeper insights. Also, LLMs are typically not used with a mathematical model for guiding the explanations, as proposed in this work, or use a user context for personalizing results.

2.5. Connection Discovery

Our approach applies a neuro-symbolic framework that combines elements of knowledge-based systems with the power of LLMs, augmented by a novel mathematical model to guide both the selection of connections and the generation of their explanations. This methodology was designed to enhance the process of relational exploration, to enable users to discover and understand complex, non-obvious connections within a knowledge graph in a personalized way. Unlike approaches focused on tasks such as object classification, our goal is to provide a comprehensive system to generate explanations based on the relationships between entities, rather than focusing only on the properties of each entity. As depicted in Figure 1, this process was broken down into three main stages, each building upon the previous one: Stage 1: Connection Discovery; Stage 2: Interestingness Scoring; and Stage 3: Explanation Generation.

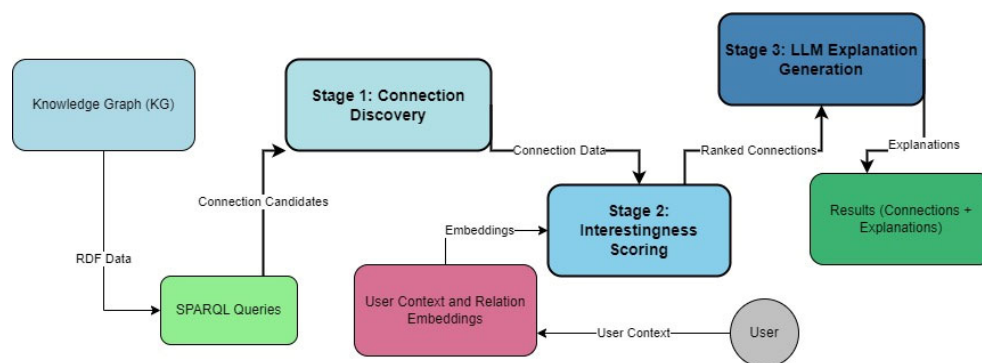


Figure 1. System architecture overview.

This initial stage (Stage 1: Connection Discovery) is dedicated to identifying potential relationship instances between entities within the knowledge graph. This step is not focused on extracting all possible relations between all entities, but rather to extract instances of relations that match predefined patterns of connections between entities that are commonly found in the domain. This helps to prune down the search space and avoid the problem of a combinatorial explosion of results. To achieve this, we utilize a series of SPARQL CONSTRUCT queries, which leverage the semantic structure of the KG to identify relationship patterns. Instead of producing explanation instances directly at this step, the SPARQL queries generate candidate connections, which can be analyzed using our mathematical model. In addition, Algorithm 1 and Figure 2 below detail the pseudo-code and steps for discovering connections between entities.

Specifically, we have identified several common types of relationships between entities, which are encoded in our set of SPARQL queries. For example:

- * "Person X was born in place Y";
- * "Person X works in place Y";
- * "Person X wrote a book about place Y";
- * "Person X created a painting that depicts place Y".

Algorithm 1 Connection Discovery

Input:

- ◇ KG: The knowledge graph.
- ◇ SPARQL_Queries: A set of SPARQL CONSTRUCT queries defining relationship patterns.

Output: Connection_Candidates CC: A set of candidate relationship instances.

1. CC = < > an empty list.
 2. For each SPARQL_Query in SPARQL_Queries:
 - Execute the SPARQL_Query on the KG.
 - For each result found by the SPARQL_Query:
 - Create a connection_instance object including:
 - * The identifier of the two connected entities
 - * The type of relationship between them
 - * Relevant metadata such as times or additional information
 - * A simple textual label for the relationship
 - Add connection_instance to Connection_Candidates.
 3. Return CC.
-

The results of the queries are used to produce sets of data that contain all the basic information required to compute our interestingness score. Each connection instance will contain:

- * entity1_id: The identifier of the first connected entity;
- * entity2_id: The identifier of the second connected entity;
- * relationship_type: The type of relationship connecting the entities;
- * relevant_metadata: Relevant metadata, such as times or additional information (when available) ;
- * explanation_text: A simple textual label describing the relationship—this will later be used by the LLM for generating explanations.

This stage, therefore, provides a foundation for the next stage (Stage 2: Interestingness Scoring) by generating a set of candidate relationships to explore. While traditional metrics like semantic relatedness (SR) and relevance capture intrinsic connections and filter irrelevant data, they often fail to identify relationships genuinely surprising, insightful, or valuable to a specific user. To address this, our design emphasizes personalized exploration through an ‘interestingness’ metric, incorporating both SR (to guide the model towards semantic connections) and contextual relevance (CR) to ensure identified relationships are not only connected but also personally relevant and engaging, pushing beyond the obvious and highlighting novel, insightful connections within the user’s profile and the overall knowledge graph.

The core of our approach lies in our novel mathematical model for quantifying the *interestingness* of a given relationship instance. We do so by combining the notions of semantic relatedness within the knowledge graph, and the contextual relevance of the connection to a specific user.

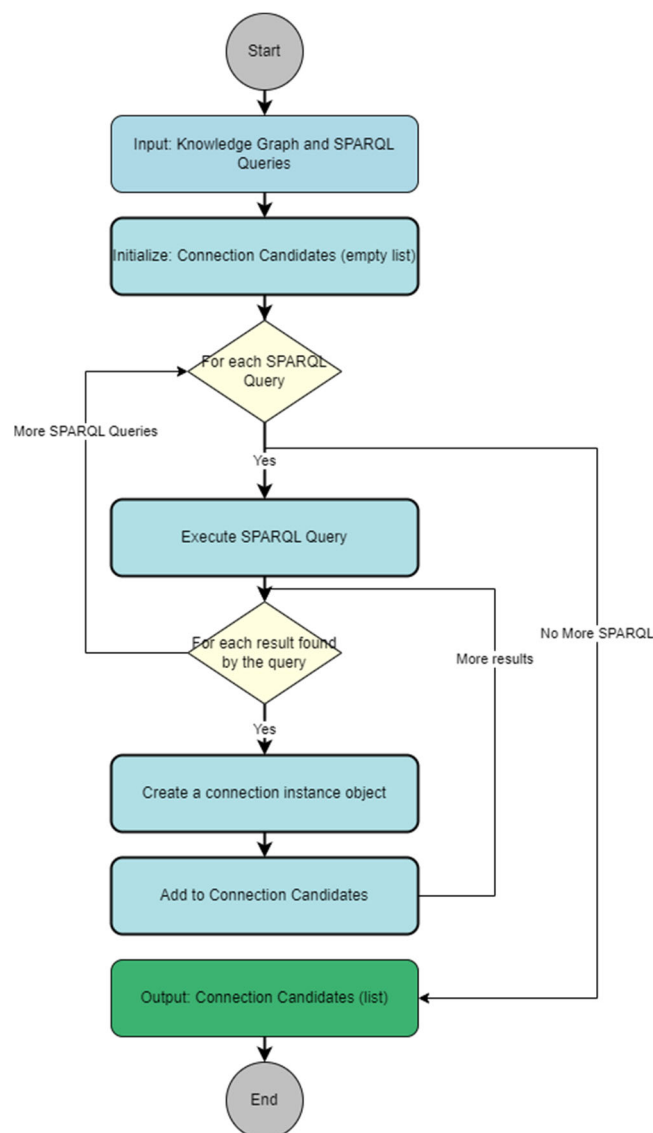


Figure 2. Connection discovery process.

As depicted in Figure 3, our ‘interestingness’ metric, $I(r)$, combines semantic relatedness, $SR(e1, e2)$, and contextual relevance, $CR(r, U)$, to prioritize relationships that are not only intrinsically connected but also aligned with the user’s specific needs and interests. We chose to incorporate both SR and CR because SR captures the inherent connections within the graph, while CR ensures the discovered relationships are relevant to the user’s unique context. Accordingly, $I(r)$ is given by Equation (1):

$$I(r) = \alpha * SR(e1, e2) + (1 - \alpha) * CR(r, U) \quad (1)$$

where:

- * $I(r)$: The overall interestingness score of the relationship instance r between entities $e1$ and $e2$;
- * $SR(e1, e2)$: The semantic relatedness score between entities $e1$ and $e2$ —this quantifies how closely related two entities are in the context of the graph structure;
- * $CR(r, U)$: The contextual relevance score of the relationship r to the user’s context U —this takes into consideration the user’s preferences, history, and expertise;
- * α : A weight parameter ($0 \leq \alpha \leq 1$) that balances the influence of semantic relatedness and contextual relevance.

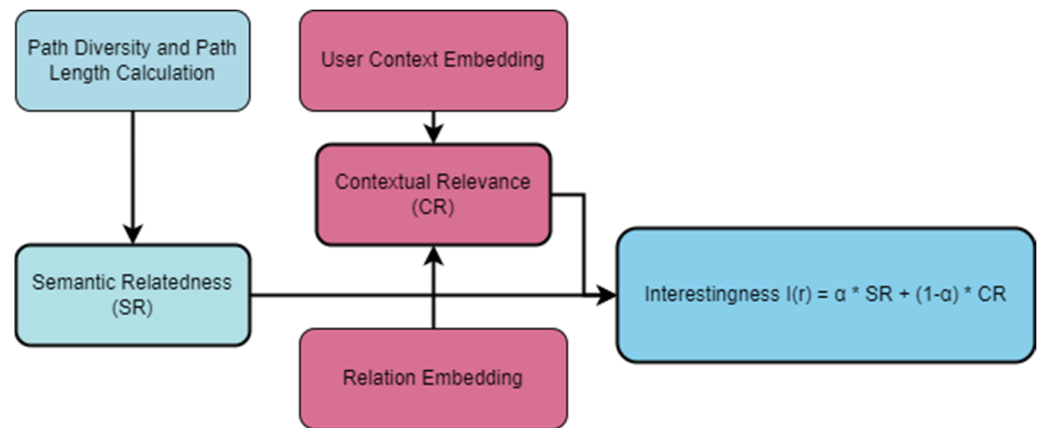


Figure 3. Interestingness scoring calculation.

The parameter α is used to control the balance between semantic relatedness and the user context. A value of α close to 1 will give more weight to the semantic relatedness measure, and a value closer to 0 will give more weight to the contextual relevance. The parameter α can be customized by the user, or it can be automatically set by the system by using a user profile.

2.5.1. Semantic Relatedness (SR)

Semantic relatedness measures the intrinsic relationship between two entities based on the connectivity and diversity of paths linking them in the KG. Our formulation favors shorter, more diverse paths, which often indicate a more meaningful semantic connection. The semantic relatedness score is calculated using Equation (2) as follows:

$$SR(e1, e2) = (1/(|P| + 1)) * \sum (1/dist(pi)) \quad (2)$$

where:

- * P : The set of all simple paths connecting entity $e1$ and $e2$ within the KG—a simple path is a path that does not visit a node twice;
- * $|P|$: The total number of simple paths connecting $e1$ and $e2$;
- * $dist(pi)$: The length of the i -th path pi , measured as the number of edges traversed.

The term $(1/(|P| + 1))$ is a normalization factor that takes into account path diversity. The higher the path diversity (the more paths are available), the lower the semantic relatedness score will be.

2.5.2. Contextual Relevance (CR)

Contextual Relevance captures how well a given relationship aligns with a user's current needs and preferences. We use a similarity measure between vector embeddings of both the relationship and the user's context. The contextual relevance is computed using Equation 3 as the cosine similarity between these embeddings:

$$CR(r, U) = \text{cosine}(v(r), v(U)) \quad (3)$$

where:

- * $v(r)$: The vector embedding of the textual description of the relationship r , generated by a Large Language Model (LLM).
- * $v(U)$: The vector embedding of the user's context U , also generated by an LLM. The user context U can be a concatenation of different types of information: search history,

domain expertise, specified user interests or other available information. The LLM can take this combined context information to generate the embeddings.

- * $\text{cosine}(v(r), v(U))$: The cosine similarity between the embeddings of $v(r)$ and $v(U)$. The score ranges from -1 to 1 , with values closer to 1 indicating a higher degree of contextual relevance.

This step introduces the use of LLMs in the system, by allowing the system to process natural language and create suitable vector embeddings.

2.6. Explanation Generation with LLMs

Once the *interestingness* score has been computed, we use an LLM (Stage 3: Explanation Generation) to generate a human-readable natural language explanation. The LLM is prompted with a structured set of information about the connection instances, which includes:

- * `entity1_description`: A concise description of the first entity;
- * `entity2_description`: A concise description of the second entity;
- * `relationship_type`: The type of relationship connecting the two entities;
- * `interestingness_score`: The value of $I(r)$;
- * `user_context_description`: A textual representation of the user's context U .

As shown in Figure 4, the following prompt structure is used to generate explanations, which is given to the LLM:

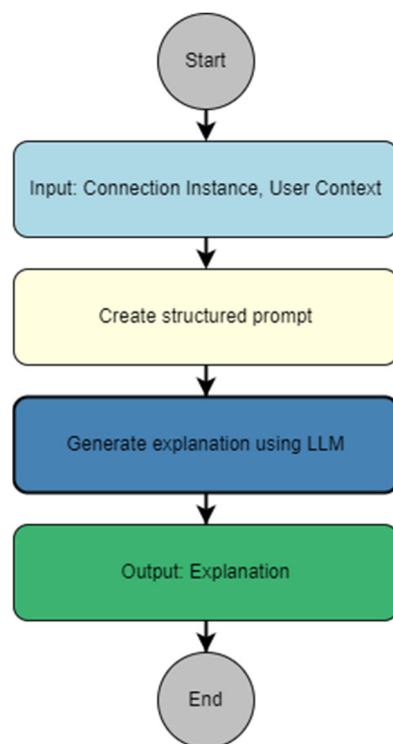


Figure 4. Explanation generation process.

“Generate a natural language explanation that connects ‘entity1_description’ and ‘entity2_description’.

The relationship type is ‘relationship_type’.

The interestingness score is: ‘interestingness_score’.

The user context is: ‘user_context_description’.

Explain this relationship in a way that reflects its interestingness, and the user context. Be specific, and avoid generic statements that could apply to other entities.”

The LLM uses the provided information to create an explanation that reflects the computed interestingness score and the user-specific context. The LLM is explicitly prompted to create a personalized explanation that highlights the nuances and significance of the relationship. The LLM is explicitly instructed to create non-generic explanations, and to use any other relevant information that may improve the quality of the final explanation. We also provide different types of examples to the LLM in order to facilitate and guide it towards creating high-quality explanations as described in Algorithm 2 below.

Algorithm 2 Explanation Generation

Input:

- ϕ Connection_Candidates: A set of candidate relationship instances.
- ϕ User Context U : A user context vector.
- ϕ LLM: The Large Language Model.

Output: Connection_Candidates CC : The same set of connections, but with the addition of natural language explanation for each instance.

1. For each connection instance r in Connection_Candidates:
 - Compute $SR(e1, e2)$ with the KG, using formula (Equation (2)) (where $e1$ and $e2$ are the entities linked by r).
 - Compute $v(r)$ and $v(U)$ vector embeddings using the LLM.
 - Compute $CR(r, U)$ using $\cosine(v(r), v(U))$ (Equation (3)).
 - Compute the $I(r)$ using $\alpha * SR(e1, e2) + (1 - \alpha) * CR(r, U)$ (Equation (1)).
 - Prompt the LLM with the following instruction:
 “Generate a natural language explanation that connects entity1_description and entity2_description. The relationship type is relationship_type. The interestingness score is: interestingness_score. The user context is: user_context_description. Explain this relationship in a way that reflects its interestingness, and the user context. Be specific, and avoid generic statements that could apply to other entities.”
 - Add the generated LLM explanation to the connection_instance.
 2. Sort the connections based on the interestingness score $I(r)$
 3. Take the top k connections, as specified by the user
 4. Return CC (top k), with the explanations.
-

This three-stage process combines the precision of knowledge-based methods with the flexibility of LLMs and the formal rigor of our mathematical model. This approach allows to go beyond simple retrieval and exploration, and allows to generate personalized explanations, based on a user’s specific needs.

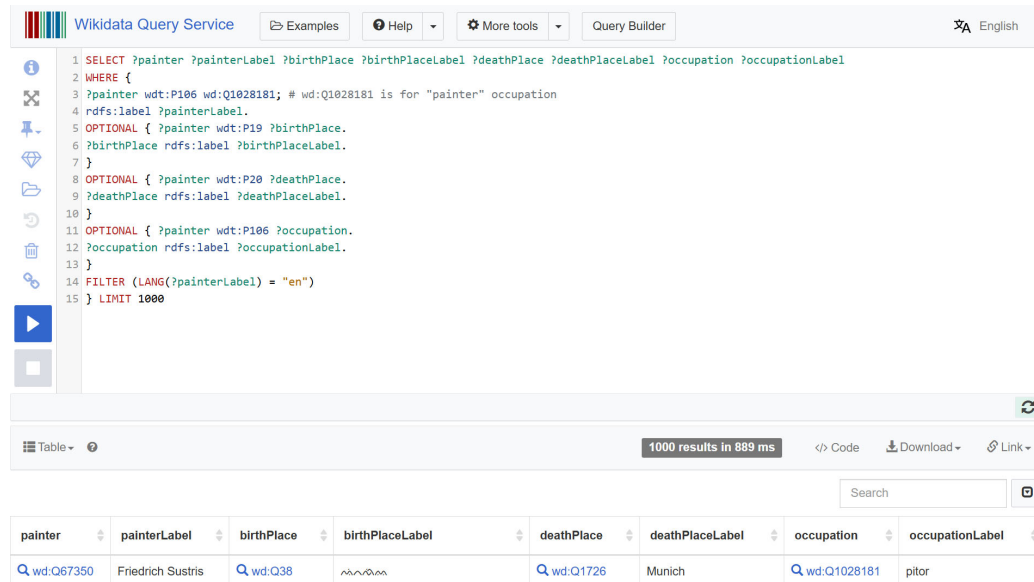
3. Results

This section details the experimental setup and experiments conducted to evaluate our neuro-symbolic framework for relational exploration, employing the public Wikidata Cultural Heritage Linked Open Data (WCH-LOD) dataset. We describe the data selection process, our implementation choices, the baseline methods used for comparison, and the evaluation metrics we have used.

3.1. Dataset

We utilized the Wikidata Cultural Heritage Linked Open Data (WCH-LOD) dataset, a publicly available RDF knowledge graph accessible at <https://query.wikidata.org/> (accessed on 5 December 2024). This dataset is specifically curated for cultural heritage

research and contains a wide range of entities, including historical figures, places, events, and artworks, along with their semantic relationships. Due to the large scale of the full dataset (comprising over 135 million triples), we focused our evaluation on a specific, yet representative, subset of the data. Figure 5 depicts a screenshot for the Wikidata Query Service with a sample SPARQL query that we used for acquiring respective data.



The screenshot shows the Wikidata Query Service interface. At the top, there are navigation links for Examples, Help, More tools, and Query Builder. The main area contains a SPARQL query:

```

1 SELECT ?painter ?painterLabel ?birthPlace ?birthPlaceLabel ?deathPlace ?deathPlaceLabel ?occupation ?occupationLabel
2 WHERE {
3 ?painter wdt:P106 wd:Q1028181; # wd:Q1028181 is for "painter" occupation
4 rdfs:label ?painterLabel.
5 OPTIONAL { ?painter wdt:P19 ?birthPlace.
6 ?birthPlace rdfs:label ?birthPlaceLabel.
7 }
8 OPTIONAL { ?painter wdt:P20 ?deathPlace.
9 ?deathPlace rdfs:label ?deathPlaceLabel.
10 }
11 OPTIONAL { ?painter wdt:P106 ?occupation.
12 ?occupation rdfs:label ?occupationLabel.
13 }
14 FILTER (LANG(?painterLabel) = "en")
15 } LIMIT 1000

```

Below the query, the results are displayed in a table format. The table has 8 columns: painter, painterLabel, birthPlace, birthPlaceLabel, deathPlace, deathPlaceLabel, occupation, and occupationLabel. The first row of results is:

painter	painterLabel	birthPlace	birthPlaceLabel	deathPlace	deathPlaceLabel	occupation	occupationLabel
wd:Q67350	Friedrich Sustris	wd:Q38	~~~~~	wd:Q1726	Munich	wd:Q1028181	pitör

Figure 5. A sample SPARQL query using the Wikidata Query Service.

We specifically targeted entities and relationships that were directly or indirectly connected to five key concepts: “Paintings” (wd:Q3305213), “Painters” (wd:Q1028181), “Museums” (wd:Q33506), “Places” (wd:Q173557), and “Historical Events” (wd:Q1190554). These concepts were chosen because they represent a diverse range of interconnected entities commonly found in cultural heritage knowledge, and provide a useful and representative set of relations. We included all entities linked to these concepts via properties such as instance of (wdt:P31), location (wdt:P276), creator (wdt:P170), and depicts (wdt:P180). We excluded triples that were not specific to these selected entities. The selection was based on a combination of semantic relevance, and common relations found in the cultural heritage domain, to capture different types of relationships. The following types of relationships were considered: “person X was born in place Y”, “person X works in place Y”, “person X wrote a book about place Y”, and “person X created a painting that depicts place Y”. This resulted in the creation of a subset that contains 105,000 entities and 500,000 relationships. The data were preprocessed using a series of steps to ensure its quality and consistency:

- * Triple Filtering: Triples that were not directly or indirectly related to the selected entity types were excluded. This ensured a cleaner and more relevant subset for evaluation.
- * Normalization: We normalized all the entities and relationships by removing blank nodes, and resolving duplicate entities. This helped in avoiding errors due to inconsistent identifiers and ensured consistent results.
- * Missing Value Handling: Incomplete entities and triples that did not include all required attributes (like entity ID and relation type) were removed.

The specific SPARQL queries used for data extraction are as follows (Box 1); noting that no data splits were created, we used the whole dataset for this experiment:

Box 1. A sample of the SPARQL queries used for data extraction.

```
# Query for extracting painter data
SELECT ?painter ?painterLabel ?birthPlace ?birthPlaceLabel ?deathPlace ?deathPlaceLabel ?oc-
cupation ?occupationLabel
WHERE {
?painter wdt:P106 wd:Q1028181; # wd:Q1028181 is for "painter" occupation
rdfs:label ?painterLabel.
OPTIONAL { ?painter wdt:P19 ?birthPlace.
?birthPlace rdfs:label ?birthPlaceLabel.
}
OPTIONAL { ?painter wdt:P20 ?deathPlace.
?deathPlace rdfs:label ?deathPlaceLabel.
}
OPTIONAL { ?painter wdt:P106 ?occupation.
?occupation rdfs:label ?occupationLabel.
}
FILTER (LANG(?painterLabel) = "en")
} LIMIT 1000

#Query for extracting painting data
SELECT ?painting ?paintingLabel ?creator ?creatorLabel ?depicts ?depictsLabel
WHERE {
?painting wdt:P31 wd:Q3305213; # wd:Q3305213 is for "painting"
rdfs:label ?paintingLabel.
OPTIONAL {?painting wdt:P170 ?creator.
?creator rdfs:label ?creatorLabel.
}
OPTIONAL{ ?painting wdt:P180 ?depicts.
?depicts rdfs:label ?depictsLabel.
}
FILTER (LANG(?paintingLabel) = "en")
} LIMIT 1000
#Similar queries were used to extract other types of entities
```

The dataset is available in RDF, and accessible through the publicly available SPARQL endpoint. This allows us to easily query the data and extract specific information. In Figure 6, we provide a sample SPARQL query and its corresponding RDF graph.

3.2. Implementation Details

The following section details our implementation of the system components:

- * LLM: We used the Llama-2-7B model, specifically the version that was released on 18 July 2023. The model was fine-tuned using a combined dataset consisting of 70% domain-specific data from the WCH-LOD dataset, and 30% synthetic data generated using our mathematical model by creating varied combinations of entities and relationships. The fine-tuning process used 10 training epochs with a batch size of 16 and a learning rate of 2×10^{-5} , and a learning rate decay of 0.01. We used AdamW as the optimizer, which provides a good balance between quality and computational efficiency. This model was chosen due to the balance between its performance and its open-source availability.
- * Embedding Model: We used the all-mpnet-base-v2 model from the Sentence Transformers library (version 2.2.2). This model was selected due to its proven ability to capture semantic similarity within textual data, which is critical for accurately mea-

asuring the relevance of user context and the relationship. The Sentence Transformers library provides an easy method to generate embeddings, and this model is known to be suitable for many different types of text.

- * Graph Database: Neo4j v4.4.14 was used as our graph database, chosen for its scalability, expressiveness, and powerful Cypher query language. We loaded the extracted RDF data into Neo4j by transforming the RDF triples into Neo4j nodes and relationships. We created indexes on the entity IDs for faster lookup. We used the cypher query language to implement our queries, which is a very powerful query language for graph databases.
- * SPARQL Engine: The Apache Jena Fuseki SPARQL server (version 4.10.0) was used as our SPARQL engine. This was chosen due to its stability, its open-source nature, and the ease of creating and executing SPARQL queries using this engine.
- * Faceted Search: We implemented a custom faceted search interface using Python Flask (version 2.3.3) and JavaScript. This implementation allows users to interactively explore the results and modify the alpha parameter using a slider, which dynamically adjusts the interestingness score to explore different perspectives on the data. The faceted search also shows summaries of the results, allowing the user to perform a more exploratory search.

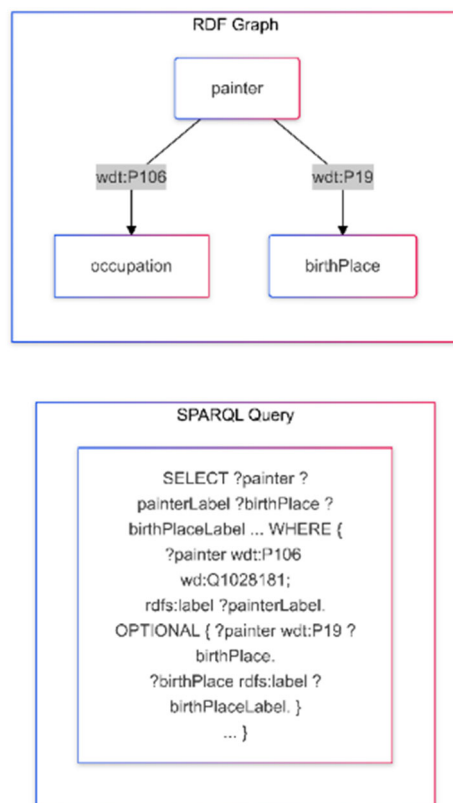


Figure 6. A sample SPARQL query and corresponding RDF graph.

3.3. Experimental Procedures

To evaluate the effectiveness of the system, we conducted two types of experiments, i.e., baseline comparisons and quantitative evaluation, since the focus of our paper is not on qualitative evaluations.

1. Baseline Comparisons: Our system was compared to two baseline methods:
 - * Graph Traversal Baseline: We implemented a standard Breadth-First Search (BFS) algorithm using Neo4j's graph database library and the shortest path method,

which provides the shortest path between two entities based on the number of edges. We used the default implementation of Dijkstra’s algorithm provided by the Neo4j library to calculate the shortest path. For the BFS, the explanation was generated by a simple textual description of the traversed path, which does not include any domain knowledge, and does not take user context into account.

- * Knowledge-Based Baseline: We created a knowledge-based baseline by manually constructing 50 SPARQL queries, and 50 simple text-based templates for generating explanations. The SPARQL queries were created based on the most frequent types of relationships found in the dataset. The templates were created using a simple keyword-based approach, and did not include a mathematical model for guiding the search. Examples of the manual SPARQL queries used for the knowledge-based baseline are as follows (Box 2):

Box 2. A sample of the manual SPARQL queries used for the knowledge-based baseline.

```
#Example query for relations about paintings
SELECT ?painting ?paintingLabel ?creator ?creatorLabel
WHERE {
?painting wdt:P31 wd:Q3305213;
rdfs:label ?paintingLabel.
?painting wdt:P170 ?creator.
?creator rdfs:label ?creatorLabel.
FILTER (LANG(?paintingLabel) = "en")
}

#Example query for relations about places
SELECT ?place ?placeLabel ?containedIn ?containedInLabel
WHERE {
?place wdt:P31 wd:Q173557;
rdfs:label ?placeLabel.
?place wdt:P131 ?containedIn.
?containedIn rdfs:label ?containedInLabel.
FILTER (LANG(?placeLabel) = "en")
}

#Similar queries were created for different types of relations

# Example template
"The painting ?paintingLabel was created by ?creatorLabel"
```

2. Quantitative Evaluation: To assess the quality and performance of the system, we computed the following metrics:
 - * Precision and Recall: We measured precision and recall against a manually created gold standard dataset. The dataset consists of 200 manually created relationships and their correct, human-generated explanations. Two domain experts with cultural heritage knowledge were employed to validate each result and generate the correct explanation. A result was considered correct if it either matched a result from the gold standard dataset, or was independently validated by the two experts. The inter-rater agreement between experts was 0.85, which indicates a high level of consistency in the annotation process. Any discrepancies in results or explanations were discussed and resolved in a meeting of the experts.
 - * Text Quality: Text quality was measured using BLEU (Bilingual Evaluation Understudy), ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation), and

METEOR (Metric for Evaluation of Translation with Explicit Ordering) scores. These are widely used in NLP for evaluating the quality of generated text. BLEU measures n-gram overlap between the generated text and reference texts, assessing the similarity of n-grams. ROUGE-L computes the longest common subsequence, capturing longer matching patterns. METEOR is an alignment-based metric that includes synonyms and stems, aligning word forms and measuring the degree of exact and near-matches between two texts. While these metrics are good for measuring the similarity of two texts, they may not capture the broader coherence or quality of the text.

- * Interestingness Correlation: We used Spearman’s rank correlation to measure the degree to which the scores produced by our mathematical model match a set of randomly generated scores. This is a measure of the monotonic relationship between the two variables. We generated the random ratings by assigning a random score between 0 and 1 to each relationship, and this was used as a baseline to assess the quality of our interestingness score. If the value is close to 1, it means the proposed method is better than random, and that it is capturing a specific type of relation between data.
- * Diversity: The diversity of the generated explanations was manually analyzed by two domain experts, who classified results based on the variety of topics, and the novelty of the discovered relationships. Novel results were classified as those results that were not commonly found using standard keyword searches or basic graph traversal methods, which include interesting insights.

All experiments were performed using the full subset of the WCH-LOD dataset. We ran five different experiments, varying the alpha parameter from 0 to 1, using increments of 0.25. We used each result as one run of the experiment. The other parameters were kept constant across the different experiments to be able to analyze the effect of the alpha parameter. The final results were similar across multiple runs, which demonstrates the consistency of the methods.

3.4. Quantitative Results

Table 1 summarizes the quantitative results, showing a clear performance improvement of our method over the baselines when evaluated on the WCH-LOD dataset.

Table 1. Quantitative results obtained using the WCH-LOD dataset.

Metric	Graph Baseline	Knowledge Baseline	Our Approach (LLM)
Precision (P)	0.28	0.45	0.70
Recall (R)	0.25	0.42	0.68
F1-score	0.26	0.43	0.69
BLEU Score	0.18	0.30	0.52

Table 1. Cont.

Metric	Graph Baseline	Knowledge Baseline	Our Approach (LLM)
ROUGE-L Score	0.24	0.32	0.58
METEOR Score	0.28	0.40	0.63
Interestingness Correlation	N/A	N/A	0.65

- * Precision, Recall, and F1-Score: The LLM-enhanced approach achieved a precision of 0.70, a recall of 0.68, and an F1-score of 0.69. These results significantly outperform both the graph traversal baseline (precision: 0.28, recall: 0.25, F1-score: 0.26) and the knowledge-based baseline (precision: 0.45, recall: 0.42, F1-score: 0.43). The quantitative results clearly demonstrate the efficacy of our approach in discovering relevant relationships while minimizing the discovery of irrelevant relationships. The results show a considerable gain compared to the other methods. The graph traversal performed particularly poorly, because it did not include any domain knowledge, and generated paths that were not very meaningful. The knowledge-based method did much better by relying on human-engineered SPARQL queries, but the LLM-enhanced approach improved over the baseline by using both the LLM and the interestingness metric, leading to better results.
- * Text Quality Metrics: The text generated by our method was evaluated using BLEU, ROUGE-L, and METEOR. Our method scores the best values in all the metrics with BLEU = 0.52, ROUGE-L = 0.58, and METEOR = 0.63. These results indicate that the LLM is able to produce more human-like texts when compared to the baselines. The knowledge-based baseline uses only simple templates, which results in poor text quality. The graph baseline produces no natural language outputs, and relies on a simple textual representation of the path.
- * Interestingness Correlation: The Spearman correlation between the interestingness scores produced by our method, and the random ratings was 0.65; this demonstrates that our mathematical model is significantly more correlated with relevance compared to random values and is, therefore, a useful tool to guide the search. Both the graph-based and the knowledge-based baselines have no correlation score, which means that the results are not very correlated with a relevant and meaningful score of interestingness. The interestingness correlation is an important guide for our method, and these results highlight its importance.
- * The experiments clearly demonstrate the advantage of using our proposed neuro-symbolic method when compared to traditional techniques.
- * Effectiveness of LLMs: By incorporating an LLM, we were able to generate more relevant and human-interpretable explanations than the template-based baseline or the graph traversal baseline, which shows the effectiveness of LLMs.
- * Importance of Mathematical Model: The novel mathematical model, based on the computation of both the semantic relatedness and contextual relevance, plays a key role in discovering relevant and non-trivial relationships, as highlighted by the interestingness correlation, which is significantly higher when compared to both baselines. This shows the importance of using a mathematical model to guide the system.

Quantitative Results: The results clearly demonstrate the higher precision, recall, F1-score, and text quality metrics when compared to the baselines, which shows that our system is able to improve on the state of the art.

4. Conclusions

This paper has presented a novel neuro-symbolic framework for relational search in cultural heritage knowledge graphs. By integrating LLMs for explanation generation and a novel mathematical formulation for interestingness, our approach significantly enhances traditional methods. Our methodology addresses the shortcomings of pure graph-based methods, which lack semantic understanding, and also of knowledge-based methods, which rely on predefined patterns and rules, offering a more dynamic and adaptable system. The use of a formal mathematical model allows our method to be more robust, personalized, and interpretable, when compared to the baselines. The results of

our quantitative experiments demonstrate that this approach not only improves precision and recall, but also increases the quality of the explanations, allows users to perform more efficient relational exploration, and highlights the importance of the interestingness measure. We believe this methodology sets a new standard for the next generation of relational search systems in cultural heritage and other domains. As a future extension to our current research work, we aim to develop more advanced techniques for automatically refining the interestingness score function based on user interactions and feedback. In addition, we plan to explore reinforcement learning to fine-tune the LLM for generating more personalized explanations. Further work on the scalability of the framework to larger and more complex knowledge graphs will also be considered.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available on request from the author.

Acknowledgments: In the preparation of this manuscript, the authors utilized Google's Gemini Large Language Model as a collaborative writing tool to assist in the articulation of complex ideas, the exploration of alternative phrasing, and the refinement of textual explanations. The author affirms that while Gemini was used as an aid in generating textual content, all conceptual development, research design, analysis, and the overall intellectual contribution of this work are solely attributable to the author.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Fan, T.; Wang, H. Research of Chinese intangible cultural heritage knowledge graph construction and attribute value extraction with graph attention network. *Inf. Process. Manag.* **2022**, *59*, 102753. [[CrossRef](#)]
2. Maree, M.; Rattrout, A.; Altawil, M.; Belkhatir, M. Multi-modality Search and Recommendation on Palestinian Cultural Heritage Based on the Holy-Land Ontology and Extrinsic Semantic Resources. *J. Comput. Cult. Herit.* **2021**, *14*, 29. [[CrossRef](#)]
3. Pellegrino, M.A.; Scarano, V.; Spagnuolo, C. Move cultural heritage knowledge graphs in everyone's pocket. *Semant. Web* **2023**, *14*, 323–359. [[CrossRef](#)]
4. Carriero, V.A.; Gangemi, A.; Mancinelli, M.L.; Marinucci, L.; Nuzzolese, A.G.; Presutti, V.; Veninata, C. ArCo: The Italian cultural heritage knowledge graph. In Proceedings of the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, 26–30 October 2019; Proceedings, Part II 18; Springer: Berlin/Heidelberg, Germany, 2019.
5. Carriero, V.A.; Gangemi, A.; Mancinelli, M.L.; Nuzzolese, A.G.; Presutti, V.; Veninata, C. Pattern-based design applied to cultural heritage knowledge graphs. *Semant. Web* **2021**, *12*, 313–357. [[CrossRef](#)]
6. Dhaliwal, C.K.; Brar, T.P.S. *Unveiling Cultural Treasures Through Digital Learning, in Digital Cultural Heritage*; CRC Press: Boca Raton, FL, USA, 2024; pp. 123–154.
7. Rahaman, H. Digital heritage interpretation: A conceptual framework. *Digit. Creat.* **2018**, *29*, 208–234. [[CrossRef](#)]
8. Shao, S.; Ribeiro, P.H.; Ramirez, C.M.; Moore, J.H. A review of feature selection strategies utilizing graph data structures and Knowledge Graphs. *Brief. Bioinform.* **2024**, *25*, bbae521. [[CrossRef](#)]
9. Capodiferro, C.; De Maria, M.; Mazzei, M.; Spreafico, M.; Bik, O.V.; Palma, A.L.; Solovyeva, A.V. Cultural Itineraries Generated by Smart Data on the Web. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 47. [[CrossRef](#)]
10. Righetto, L.; Khademizadeh, M.; Giachetti, A.; Ponchio, F.; Gigilashvili, D.; Bettio, F.; Gobbetti, E. Efficient and user-friendly visualization of neural relightable images for cultural heritage applications. *ACM J. Comput. Cult. Herit.* **2024**, *17*, 1–24. [[CrossRef](#)]
11. Tartari, G.; Hogan, A. *WiSP: Weighted Shortest Paths for RDF Graphs*; Universidad de Chile: Santiago, Chile, 2018.
12. Hyvönen, E. Using the Semantic Web in digital humanities: Shift from data publishing to data-analysis and serendipitous knowledge discovery. *Semant. Web* **2020**, *11*, 187–193. [[CrossRef](#)]
13. Liang, K.; Liu, H.; Shan, M.; Zhao, J.; Li, X.; Zhou, L. Enhancing scenic recommendation and tour route personalization in tourism using ugc text mining. *Appl. Intell.* **2024**, *54*, 1063–1098. [[CrossRef](#)]
14. Maree, M.; Belkhatir, M. Addressing semantic heterogeneity through multiple knowledge base assisted merging of domain-specific ontologies. *Knowl.-Based Syst.* **2015**, *73*, 199–211. [[CrossRef](#)]

15. Mountantonakis, M.; Tzitzikas, Y. Generating SPARQL Queries over CIDOC-CRM using a Two-Stage Ontology Path Patterns Method in LLM Prompts. *ACM J. Comput. Cult. Herit.* **2025**, *18*, 1–20. [[CrossRef](#)]
16. Cheng, G.; Shao, F.; Qu, Y. An empirical evaluation of techniques for ranking semantic associations. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2388–2401. [[CrossRef](#)]
17. Cheng, G.; Zhang, Y.; Qu, Y. Explass: Exploring associations between entities via top-k ontological patterns and facets. In Proceedings of the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, 19–23 October 2014; Proceedings, Part II 13; Springer: Berlin/Heidelberg, Germany, 2014.
18. Rantala, H.; Leskinen, P.; Peura, L.; Hyvönen, E. Representing and searching associations in cultural heritage knowledge graphs using faceted search. In *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI*; IOS Press: Amsterdam, The Netherlands, 2024; pp. 420–435.
19. Ismail, S.; Shaikh, T. EnSense—A commonality checker for Semantic Web. In Proceedings of the 2017 International Conference on IoT in Social, Mobile, Analytics and Cloud (I-SMAC), Palladam, India, 10–11 February 2017; IEEE: Piscataway, NJ, USA, 2017.
20. Li, J.; Bikakis, A. Towards a Semantics-Based Recommendation System for Cultural Heritage Collections. *Appl. Sci.* **2023**, *13*, 8907. [[CrossRef](#)]
21. Bugueño, M.; Biswas, R.; de Melo, G. *Graph-Based Explainable AI: A Comprehensive Survey*; HAL Open Science: Lyon, France, 2024.
22. Bobasheva, A.; Gandon, F.; Precioso, F. Learning and reasoning for cultural metadata quality: Coupling symbolic AI and machine learning over a semantic web knowledge graph to support museum curators in improving the quality of cultural metadata and information retrieval. *J. Comput. Cult. Herit. JOCCH* **2022**, *15*, 1–23. [[CrossRef](#)]
23. Wang, R.; Deng, J.; Guan, X.; He, Y. A framework of genealogy knowledge reasoning and visualization based on a knowledge graph. *Library Hi Tech* **2024**, *42*, 1977–1999. [[CrossRef](#)]
24. Wang, Y.; Liu, J.; Wang, W.; Chen, J.; Yang, X.; Sang, L.; Wen, Z.; Peng, Q. Construction of Cultural Heritage Knowledge Graph Based on Graph Attention Neural Network. *Appl. Sci.* **2024**, *14*, 8231. [[CrossRef](#)]
25. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [[CrossRef](#)]
26. Díaz-Rodríguez, N.; Pisoni, G. Accessible cultural heritage through explainable artificial intelligence. In Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, Genoa, Italy, 12–18 July 2020.
27. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.