College of Computer Science and Technology, Jilin University

– NULL Group –

Construction of Floral Knowledge Graph with Data Integration

Document Data:	Reference Persons:
- 2023-11-07 -	- ChenYu Li, YongZhen Li, YunXia Zhang -

Contents

Revision History:

Revision	Date	Author	Description of Changes

1 Team Members and Distribution

Table 1 provides an overview of the Team Members and their respective Distribution.

Table 1: Team Members

ID	Name	Role
2023532027	Chenyu Li	Knowledge Engineer
2023534036	Yongzhen Li	Data Scientist
2023532106	Yunxia Zhang	Project Manager

2 Introduction

2.1 Introduction to Knowledge Graph

Knowledge Graph is a graphical data model used for organizing, storing, and representing knowledge. It consists of entities and relationships between entities, typically represented in a graph where entities are nodes and relationships are edges. Key concepts about Knowledge Graph include:

- Knowledge Representation and Organization: Knowledge Graph is a data structure used to represent and organize complex knowledge. It can include information from various domains, such as individuals, locations, events, concepts, facts, relationships, and more;
- Entities and Relationships: The fundamental elements in a Knowledge Graph are entities and relationships. Entities represent objects, concepts, or entities in the real world, while relationships denote connections or associations between different entities;
- Graph Structure: Knowledge Graph adopts a graph structure where entities are depicted as nodes, and relationships are depicted as edges. This graphical representation facilitates analysis and visualization, helping people better understand the relationships between pieces of knowledge;
- Semantic Associations: Knowledge Graphs can contain semantic information related to entities and relationships, which aids in explaining the meaning of entities and the underlying significance of relationships;
- Cross-Domain Applications: Knowledge Graphs can span a wide range of domains, from natural language processing and bioinformatics to business intelligence and cultural heritage preservation. Knowledge Graphs from different domains can be interconnected to form larger-scale knowledge graphs;
- Automated Knowledge Extraction: Building Knowledge Graphs typically involves automated knowledge extraction and annotation to gather knowledge from sources like text, databases, and the internet. This contributes to increasing the scale and update frequency of Knowledge Graphs;
- Question-Answering and Reasoning: Knowledge Graphs can be utilized in question-answering systems and knowledge reasoning. They can answer user queries, perform complex searches, and support reasoning engines to discover new knowledge;

- Standards and Formats: Knowledge Graphs often adhere to specific standards and formats such as RDF (Resource Description Framework) and OWL (Web Ontology Language) to ensure data interoperability and scalability.

Applications: Knowledge Graphs are utilized in various domains, including search engines, social media, healthcare, the Internet of Things (IoT), intelligent assistants, recommendation systems, and more. They are employed to enhance tasks like information retrieval, personalized recommendations, semantic search, and more. In summary, Knowledge Graphs are a powerful tool for organizing and representing knowledge, facilitating cross-domain information integration and knowledge management. The concept of Knowledge Graphs has found widespread applications in multiple fields, providing robust tools for problem-solving, decision support, and improving user experiences.

2.2 Introduction to Data Alignment

Data Alignment in Knowledge Graphs refers to the coordination and integration of data from different sources, formats, or domains in order to ensure consistency, comparability, and interoperability among the data. It is a crucial task in the construction and maintenance of Knowledge Graphs, aimed at addressing data heterogeneity and diversity to achieve improved data integration and knowledge discovery. Here are detailed concepts related to data alignment in Knowledge Graphs:

- Data Heterogeneity: Different data sources may use various formats, schemas, terminology, and standards, leading to data heterogeneity. The goal of data alignment is to resolve these heterogeneities, enabling data to work together within a unified framework;
- Data Integration: One of the primary objectives of data alignment is to consolidate scattered data into a unified Knowledge Graph. This can include structured data, unstructured data, semi-structured data, and more, to achieve comprehensive knowledge representation;
- Relationship Establishment: Establishing relationships between different data sources is crucial in the data alignment process. This can involve relationships between entities, mappings between attributes, or identifying the same entities across different data sources;
- Data Preprocessing: Data alignment often requires data preprocessing, including data cleaning, deduplication, normalization, and standardization to ensure data quality and consistency;
- Mapping Rules: To achieve data alignment, mapping rules or transformation rules need to be defined to map data from different sources into a common format and schema. This includes attribute mapping, entity identification mapping, and more;
- Semantic Consistency: Data alignment is not just about aligning data structures; it also involves considering semantic consistency. This includes ensuring that terms and concepts from different data sources can correctly map to a common semantic model;
- Tools and Technologies: Various data alignment tools and technologies are available, including graph databases, ontology technologies, natural language processing tools, and rule engines, to support data alignment tasks;
- Application Domains: Data alignment is widely applied across multiple domains, including enterprise knowledge management, scientific research, natural language processing, intelligent search engines, social network analysis, and more;

- Ongoing Maintenance: Data alignment is not a one-time task; it requires continuous maintenance and updates to reflect changes in data sources and the growth of knowledge.

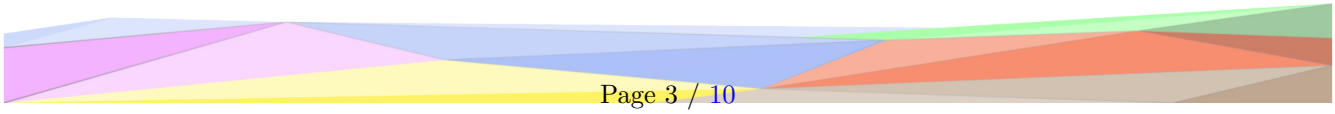
Data alignment is a crucial step in the construction of Knowledge Graphs. It helps address the challenges posed by heterogeneous data, offering a more consistent, interoperable knowledge representation that supports the integration and discovery of knowledge, thereby promoting the development of intelligent applications and analytics.

2.3 Introduction to Data Ambiguity

Data Ambiguity in Knowledge Graphs refers to the presence of multiple possible interpretations or meanings within the data, making it challenging to precisely understand the data's significance. This ambiguity can lead to misinterpretations, incorrect inferences, and inaccurate reasoning about the data. In Knowledge Graphs, data ambiguity is a common challenge, as Knowledge Graphs often consist of multiple data sources, different domains, and various contributors, which can introduce polysemy and vagueness into the data. Here are concepts related to data ambiguity in Knowledge Graphs:

- Polysemy: Polysemy in data refers to a term or concept having multiple possible meanings. For example, the name of an entity might have different interpretations in different contexts;
- Vagueness: Data vagueness occurs when the information in the data is not sufficiently clear, making it difficult to determine its exact meaning. This can result from incomplete descriptions, lack of context, or vague descriptions;
- Causes of Ambiguity: Data ambiguity can be caused by factors such as the diversity of data sources, incomplete data models, unclear terminology, multilingual data, context variations, and more;
- Context Sensitivity: Data ambiguity is often context-dependent, where the same data may have different meanings in different contexts. Therefore, resolving data ambiguity typically requires considering contextual information;
- Semantic Disambiguation: To address data ambiguity, semantic disambiguation is often necessary, which involves determining the precise meanings of terms or concepts in the data. This can be achieved through context analysis, semantic tagging, additional information in knowledge bases, and more;
- Standardization: In Knowledge Graphs, using standardized terms and models can help reduce data ambiguity. Employing common ontologies or standardized data models can provide a more consistent data representation;
- Inference and Resolution: Data ambiguity can have negative impacts on Knowledge Graph reasoning and query. Therefore, strategies to resolve data ambiguity should be considered in knowledge inference and query processes;
- User Interface and Interaction: Addressing data ambiguity is not solely a technical issue but also involves the design of user interfaces and user interactions. Clear interfaces and contextual cues can help users better understand the data.

In conclusion, data ambiguity is a significant challenge in Knowledge Graphs that requires a comprehensive consideration of factors such as data sources, context, and semantics to provide a more accurate, consistent, and understandable knowledge representation. Addressing data ambiguity contributes to improving the quality and usability of Knowledge Graphs, thereby supporting better decision-making and knowledge discovery.



3 Context

3.1 Domain of Interest

The Domain of Interest for this project revolves around the selection of suitable flowers for specific locations. The motivation behind this project stemmed from the experience of relocating to a new workstation and recognizing the need to embellish the desk with flowers. The objective was to identify flowers that possess air-purifying, formaldehyde-removing, and radiation-blocking capabilities. Given that the workstation receives ample sunlight from 8 AM to 4 PM, the question arose as to which flowers would be most suitable for this environment.

Initial attempts to find answers involved using search engines but proved challenging in obtaining comprehensive and accurate information that met the specific requirements. Existing suggestions provided limited options for flowers and lacked detailed information.

Exploration of specialized flower websites also had limitations, as they focused on specific categories of flower properties, such as air purification without radiation-blocking capabilities.

Recognizing these gaps, the project aimed to develop a comprehensive and personalized recommendation system that addresses the specific criteria for flower selection. The focus was on providing a refined solution that caters to the requirements of different locations and preferences.

The proposed idea involved inputting information about the location, specifically an office with computer usage requirements, and seeking flowers that can block radiation while offering ornamental value.

The Flower Knowledge Graph is a knowledge graph project centered around flowers as core entities. Each flower has certain attributes and features, such as name, color, fragrance, etc. They belong to different categories of flowers and have various functionalities and adaptability to different application environments.

3.2 Competency Questions

Taking into account the personas in the scenarios defined, we create Competency Questions (CQs):
Raw Competency Questions:

CQ1: What kind of flowers are suitable for a sunny new office?

CQ2: What should be noticed when cultivating flowers in a living room with insufficient sunlight?
Kernel Competency Questions:

CQ1: Flower selection, Suitability, Sunny location, New office

CQ2: Flower selection, Notice, Living room, Insufficient sunlight
Analysed Competency Questions:

CQ1:
common: Flower, Environment
core: SuitabilityFlower selection
contextual: Sunny location, New office

CQ2:
common: Flower , Environment
core: notice, Flower selection
contextual: Living room, Insufficient sunlight
Classified Competency Questions:

CQ1:
Common OBJECTFlower, Environment
Core FUNCTION: Suitability ACTION: Flower selection
Contextual FUNCTION: Sunny location, New office

CQ2:
Common OBJECTFlower, Environment
Core FUNCTION: Notice ACTION: Flower selection
Contextual FUNCTION: Living room, Insufficient sunlight

4 Entity–relationship model

Based on the Common and Core entities in Table 1, we design an Entity–relationship (ER) model as Figure 3.

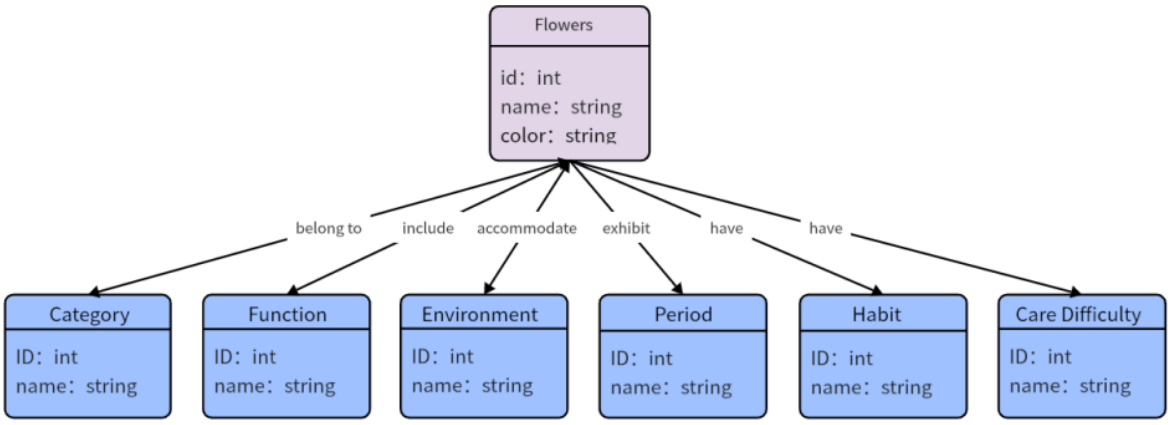


Figure 1: Entity-Relationship Diagram

Explaination of the ER:
Entity:

Flowers: As the main entities of the knowledge graph, flowers are the core entities. Each flower has certain attributes and features such as name, color, fragrance, etc. They belong to different categories of flowers and have different flower functionalities and adaptability to various application environments.

Flower Categories (12 categories): These categories can include different types of flowers such as novelty, aromatic, hydroponic, etc. A flower can belong to one or more flower categories.

Flower Functionalities (8 categories): These categories describe the functionalities of flowers, such as ornamental, medicinal, air purifying, etc. Each flower can have one or more functionalities.

Application Environments (14 categories): These categories represent the different environments in which flowers can grow and be used, such as balcony, coffee table, study room, etc. Different flowers can adapt to different application environments.

Flowering Period (6 categories): These categories describe the flowering periods of flowers, including spring, summer, autumn, winter, and non-flowering.

Habit (2 categories): These categories describe the growth habits of flowers, including shade-loving and sun-loving.

Care Difficulty (4 categories): These categories indicate the levels of care difficulty for different flowers, ranging from easy to difficult.

Relationship:

Flowers and Application Environments: There is a many-to-many relationship between flowers and application environments, which means that a flower can adapt to multiple different application environments, and an application environment can be suitable for multiple different flowers.

Flowers and Flower Functionalities: There is a one-to-many relationship between flowers and flower functionalities, where a flower can have one or more functionalities, such as ornamental and medicinal.

Flowers and Flower Categories: There is a many-to-many relationship between flowers and flower categories, where a flower can belong to multiple different flower categories. For example, lavender can belong to both the aromatic and foliage categories.

Flowers and Flowering Period: There is a one-to-one relationship between flowers and flowering periods, where each flower has a specific flowering period.

Flowers and Care Difficulty: There is also a one-to-one relationship between flowers and care difficulty, where each flower has a specific level of care difficulty.

These entities and relationships form the fundamental structure of the Flower Knowledge Graph, describing the characteristics, functionalities, and adaptability of different flowers, as well as their relationships with various environments and application scenarios. This will help users better understand and utilize knowledge related to flowers.

5 Project Description

Driven by the mentioned motivation in 3.1, a flower knowledge map was developed. Figure 2 illustrates how the knowledge graph meets the specified requirements. When the location is entered as "office", the function is specified as viewing, and the system is asked to recommend flowers that block radiation, five flowers that fulfill these criteria are returned.

```
MATCH (flower:`花卉`)-[:具有作用]→(function:`花卉功能` {name: '防辐射花卉'})
WITH flower
MATCH (flower)-[:accommodate]→(environment:`应用环境` {name: '办公室花卉'})
WITH flower
MATCH (flower)-[:具有作用]→(environment:`花卉功能` {name: '观赏花卉'})
RETURN flower limit 5;
```

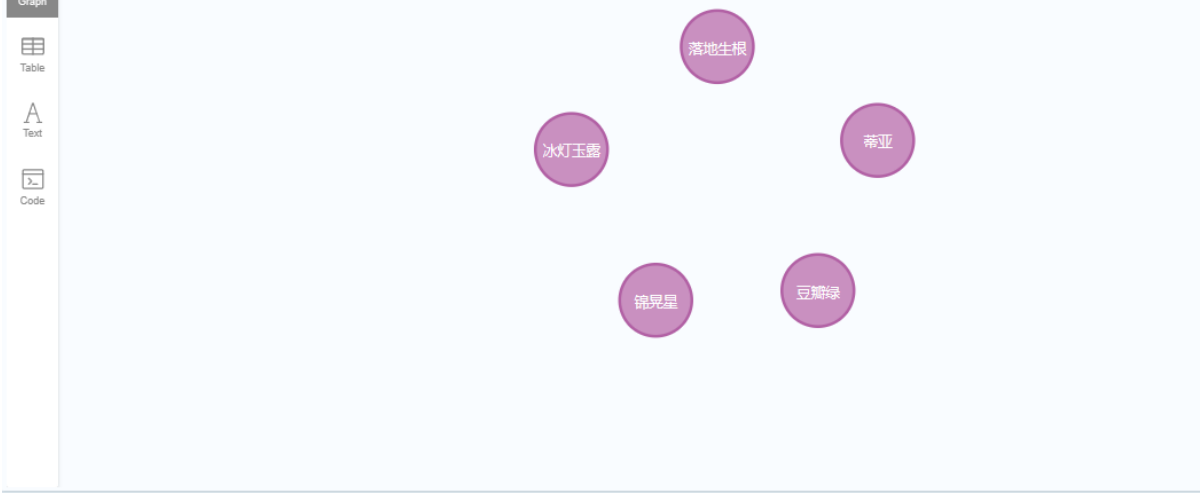


Figure 2: Illustration of how the knowledge graph aligns with the specified requirements.

Figure 3 provides specific information about the flowers. Expanding the flower node reveals that both "ice lanterns, rain and dew" and "Douban green" meet the defined requirements.

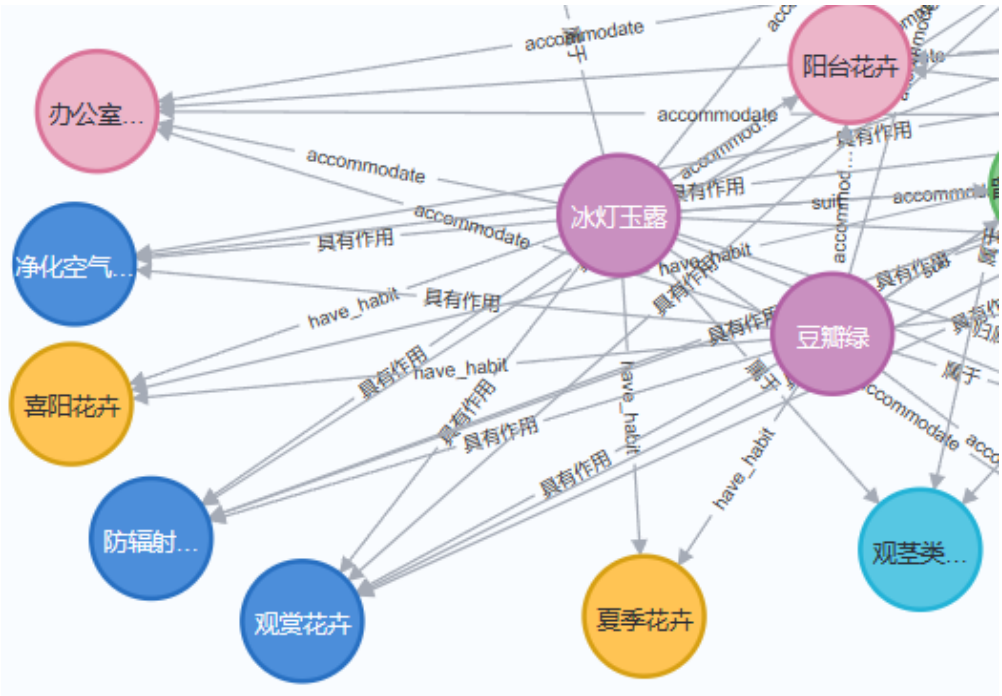


Figure 3: Illustration of flowers satisfying user requirements as retrieved by the knowledge graph search.

Figure 4 represents the final constructed knowledge graph. We constructed the knowledge graph using the latest version of Neo4j, coupled with the py2neo Python package to store data in Neo4j. We sequentially created nodes such as "Floral Encyclopedia" , "Floral Varieties" , "Flower" , "Flowering Period" , "Habit" , "Care Difficulty" , "Floral Functions" , "Floral Category" , and established relationships between entities based on the ER diagram. This successful process allowed us to build a comprehensive knowledge graph.

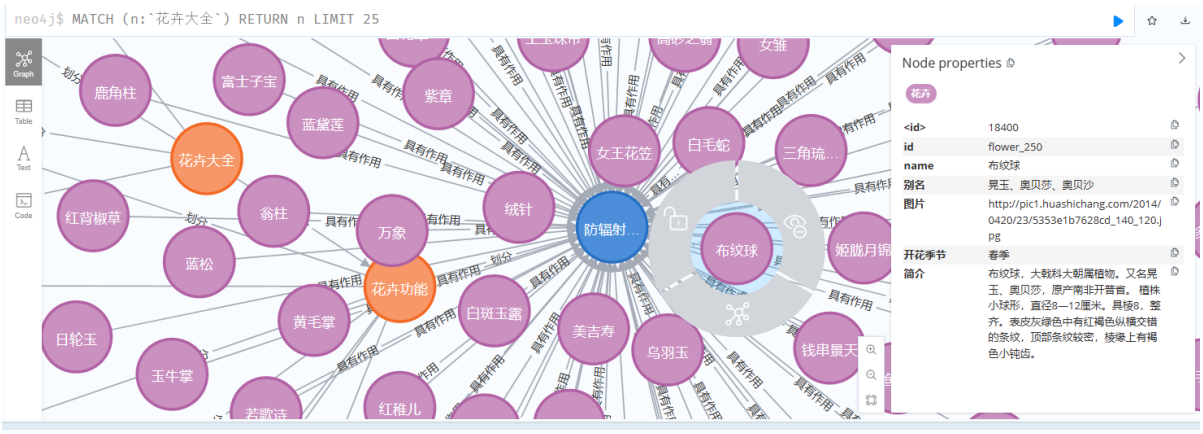


Figure 4: Final constructed knowledge graph

6 Resource Classification (with Availability)

6.1 Knowledge Resources

http://www.openkg.cn/

6.2 Data Resources

http://www.aihuhua.com/

7 Implementation of Data Alignment

Understanding Flower Language Data: Firstly, understand the sources and structure of flower language data. Flower language typically includes meanings, symbolism, and traditional uses associated with different flower varieties. Make sure you have a clear understanding of the format and content of the flower language data.

Data Preprocessing: Before aligning the flower language data with the knowledge graph, preprocessing may be required. This involves cleaning, normalizing, and structuring the flower language data to make it consistent with your knowledge graph data.

Entity Identification: Match or identify the flower varieties in the flower language data with the flower entities in your knowledge graph. This can be done based on flower names, specific attributes, etc.

Establish Associations: Establish associations between each flower variety and the flower language data. This can be done by creating a new relationship that links the flower entity with the flower language data entity.

Data Mapping: Define how the meanings, symbolism, or other attributes in the flower language data map to the knowledge graph. This may involve using mapping rules or matching the attributes of the flower language data with the attributes in the knowledge graph.

Semantic Consistency: Ensure that the terminology and meanings in the flower language data are semantically consistent with the knowledge graph. This helps reduce ambiguity.

Validation and Testing: Before aligning the flower language data with the knowledge graph, perform validation and testing to ensure the alignment is accurate and does not introduce errors.

Continuous Updates: Once the alignment is complete, ensure that your knowledge graph and flower language data are kept up to date to reflect new flower meanings or symbolism.

Querying and Presentation: Finally, ensure that users can query and access the flower language data, possibly by adding appropriate query interfaces and presentation methods.

8 Resolution of Data Ambiguity

Entity Naming Convention: Define precise and unique naming conventions for each entity. Ensure that the names of flower categories, flower functionalities, application environments, and other entities are clear, specific, and avoid ambiguous or duplicate naming.

Context Disambiguation: Provide contextual information for each entity during the construction of the knowledge graph, including descriptions, attributes, and relationships. By providing more contextual information, it can help eliminate potential ambiguities. For example, when describing a flower category, provide detailed information about its characteristics, morphology, growing conditions, etc., to differentiate it from other flower categories.

Attribute Description and Classification Standards: Provide detailed descriptions and classification standards for each entity attribute to reduce data ambiguity. Define clear features, classification criteria, and descriptions for flower categories, flower functionalities, application environments, flowering periods/habits, and care difficulty, giving them explicit meanings.

Relationship Description and Handling of Many-to-Many Relationships: For the many-to-many relationship between flowers and application environments, add relationship descriptions to explain that a flower can adapt to multiple application environments and an application environment can be suitable for multiple flowers. This helps eliminate relationship ambiguities and provides more contextual information.

Attribute Value Standardization: For attributes with diverse values, such as flowering periods and care difficulty, define a unified representation format. For example, use a specific date format to represent flowering periods or use fixed levels to categorize care difficulty, avoiding different interpretations and ambiguities.

Data Validation and Review: Validate and review the data to ensure accuracy and consistency, eliminating potential data ambiguities and errors. Additionally, continuously update and maintain the data to keep the knowledge graph accurate and practical.

References