Caring is Sharing – Exploiting the Value in Data for Health and Innovation
M. Hägglund et al. (Eds.)

doi:10.3233/SHTI230243

Building a Disease Knowledge Graph

Enayat RAJABI[a,b,1] and Somayeh KAFAIE[b,c]
[a] *Cape Breton University, Sydney, NS, Canada*
[b] *Center for Applied Intelligent Systems Research, Halmstad University, Sweden*
[c] *Saint Mary's University, Halifax, NS, Canada*
ORCiD ID: Enayat Rajabi https://orcid.org/0000-0002-9557-0043
Somaye Kafaie https://orcid.org/0000-0002-5685-6487

Abstract. Knowledge graphs have proven themselves as a robust tool in clinical applications to aid patient care and help identify treatments for new diseases. They have impacted many information retrieval systems in healthcare. In this study, we construct a disease knowledge graph using Neo4j (a knowledge graph tool) for a disease database to answer complex questions that are time-consuming and labour-intensive to be answered in the previous system. We demonstrate that new information can be inferred in a knowledge graph based on existing semantic relationships between the medical concepts and the ability to perform reasoning in the knowledge graph.

Keywords. Knowledge Graph, Disease Database, Neo4j.

1. Introduction

Many databases and websites have emerged to allow users to explore medical information on the Web and search for a specific medical disorder or disease. With the complex relationships among different disease concepts, drug ingredients and symptoms in a database, searching for information requires browsing and traversing such relationships to find the tailored answer, which is time-consuming and labour-intensive. Providing a system by defining semantic relationships between different concepts facilitates the search and leads to the discovery of new relations. Hence, many organizations construct a semantic network or knowledge graph to include information about concepts, events and relationships in a graph and perform reasoning [1]. Semantic relationships between entities in a knowledge graph create new concepts and a new level of understanding, allowing machines to make new connections between entities and build new knowledge. Knowledge graphs also have significantly impacted many AI-related applications and information retrieval systems. They have been widely used in healthcare and biomedical systems in applications such as finding relationships between diseases and recommending drugs to patients. They are also utilized in studying genomes and pharmaceutical applications to identify new properties of drugs by predicting drug-drug interactions. This process can help identify a new disease treatment.

Several disease database systems, such as DISNET [2] and DisGeNET [3], leverage semantic relationships between disease concepts, signs, drugs, symptoms and

[1] Corresponding Author: Enayat Rajabi, 1250 Grand Lake Rd, Sydney, NS, Canada; E-mail: enayat_rajabi@cbu.ca.

diagnostic tests associated with a disease to retrieve knowledge from PubMed and Wikipedia. Rather than creating a knowledge graph, extracted data in such systems is usually stored in a database. In this study, we construct a knowledge graph to visualize disease-gene associations, human disease, medications, symptoms and signs based on a cross-referenced disease database. We demonstrate the efficiency of this approach by designing some questions and providing reasoning on the constructed graph. Our results show that knowledge graph reasoning can accelerate identifying critical clinical discoveries and help infer missing facts from existing ones.

2. Method

A knowledge graph can be constructed in different steps, such as data and knowledge acquisition, knowledge enrichment, knowledge storage, and retrieval. The primary data and knowledge acquisition resources include unstructured, semi-structured or structured data. It can be performed based on entity and relation extraction. In the knowledge graph construction process, either a database is constructed or the knowledge graph is created using pre-existing databases. This is the stepping stone towards having information in a machine-readable format, laying the foundation for semantic models, such as ontologies, understanding and using the existing vocabularies, and mapping relationships to add context and meaning to different and distinct data. After mapping, the second step is visualizing the knowledge graph to access a graph-based data view. It allows us to retrieve data, explore the knowledge graph, and write our questions as queries. The following subsections will explain the disease database and the knowledge graph construction process.

2.1. Disease Database

The Diseases Database[2] provides a system for extracting disease-gene associations from biomedical abstracts. It is a cross-referenced index of human disease, medications, symptoms, signs, abnormal investigation findings, etc. This database presents a medical textbook-like index and search portal covering internal medical disorders, symptoms and signs, congenital and inherited disorders, infectious diseases and organisms' medications, common hematology, and biochemistry investigation abnormalities. The database owner provided a subset of 9,400 disease concepts (e.g., diseases, symptoms, and drugs) with 45,507 relationships (e.g., cause-effect, drug family, and risk factor) to us in a Comma Separated Values (CSV) format for data analysis and graph construction.

2.2. Graph Construction

We imported the dataset into a relational database (MySQL) and cleaned it by removing unnecessary columns (e.g., IDs). Table 1 shows a sample of records of the final created dataset before knowledge graph construction. Each disease concept in the dataset is considered a disease node denoted as n, related to another concept (m) using a relationship r. Each dataset record in the database is shown as subject (n), predicate(r),

[2] http://www.diseasesdatabase.com/ , accessed on 2023-03-12.

and object (*m*) based on the Semantic Web notation[3]. For example, "Phenprocoumon interacts with Paracetamol" is denoted as (Phenprocoumon (*n*), interacts with (*r*), Paracetamol(*m*)) in the knowledge graph. We leveraged the Neo4j Desktop software4 to construct the disease graph and define relationships between the nodes. Neo4j is a graph data platform to store, handle, and query highly connected data in a dataset. Using the LOAD CSV command, we imported the dataset into Neo4j and visualized the knowledge graph. The final disease knowledge graph had 9,400 subject nodes and 3,913 object nodes. The software inferred more than 45,000 relationships.

Table 1. A sample of disease database records

Item-1	Relation	Item-2
Ethanol	may cause	Metabolic acidosis
Amantadine	may cause	Livedo reticularis
Aflatoxins	is a risk factor for	Liver cancer
Respiratory acidosis	is a subtype of	Acidosis

2.3. Query Construction

We followed a question-answering template proposed by [4] for constructing queries with various complexity levels. We also asked a medical doctor to design three questions that require logical reasoning (e.g., hierarchical or inverse) over the subgraphs of the knowledge graph. We later converted the questions to the Neo4j Cypher queries[5] and visualized them in the software. Two types of questions (entity-based or relation-based) were defined. The entity-based question should include at least one relationship between the disease concepts. For example, "What does Dysmenorrhoea cause?" looks for the "cause" relationship between "Dysmenorrhoea" and another entity. To answer this question, an end-user should explore the disease website in different categories to identify the Dysmenorrhoea causes. The relation-based questions use numerous relations and some annotators with various logical operators such as AND, OR, and NOT. The relation between subject and object might be hierarchical or inverse in this type. For example, "Which drug from the '*Folic acid antagonists*' category may cause '*Hepatic failure*' "?

3. Results

The designed questions, the Cypher queries and the results are shown below:

- Question: What diseases are related to "Anemia"? *(Subject: Unknown, Relation: Unknown, Object: "Anemia"?*
  Cypher query: `MATCH (n:First_Item)-[:Relation{type: "may cause"}]-> (b) WHERE b.Name =~"Anemia" RETURN n, b`

3 https://www.w3.org/standards/semanticweb , accessed on 2023-03-12.

4 https://neo4j.com/ , accessed on 2023-03-12.

5 https://neo4j.com/docs/getting-started/current/cypher-intro/ , accessed on 2023-03-12.

Figure 1. The result of Query 1.

- Question: What Antibiotics drugs may cause "Hepatic failure"? *(Subject: Unknown (a subcategory of Antibiotics drugs), Relation: "may cause", Object: "Hepatic failure")*
  Cypher query:
```
MATCH (n:First_Item)-[c:Relation{type:'may cause'}]->(b:Second_Item{Name:'Hepatic failure'})
MATCH (n)-[dr:Relation{type:'belongs to the drug family of'}]-> (f{Name:'Antibiotics'}) RETURN n, c, b
```

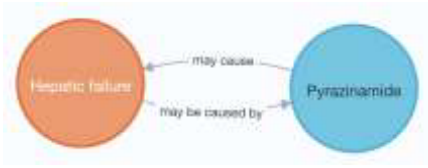

Figure 2. The result of Query 2.

- Question: What are the causes and risk factors of "Acne vulgaris"? *(Subject: Unknown, Relation: ("may cause" and "risk factor for"), Object:"Acnevulgaris").*
  Cypher query:
```
MATCH (n:First_Item)-[:Relation{type:'may cause'}]-> (b)WHERE b.Name =~ "Acne "+"(?i).*"
MATCH (f)-[dr:Relation{type:'is a risk factor for'}]->(b)RETURN n, f, b
```

Figure 3. The result of Query 3.

The results of Neo4j queries our manual search on the disease database website. To validate the results, we manually explored the disease database website and traversed the relationships between the disease items to retrieve the answers. For example, for the question: What ”Antibiotics” drugs may cause “Hepatic failure”?, we first searched the disease website for “Hepatic failure” keywords. Then, we selected the “may be caused by” relationship and browsed the entities listed there to find their categories and subcategories. If the category was “Drug” and the subcategory was “Antibiotics”, we checked the drug names with the No4j answer.

4. Discussion and Conclusion

Constructing a knowledge graph for a database allows for connecting various concepts semantically. In healthcare, knowledge graph provide a system to explore the connections and discover indirect relationships among diseases, drugs, symptoms and other entities. Our study found some relationships in the disease knowledge graph that were missing from the disease database website. For example, “Acne vulgaris” may be caused by “Propionibacterium acnes” based on the semantic relationships between these two items in the graph; however, the website does not have this information. From a practical point of view, the presented system can be improved by providing a query interface for health professionals to write questions in natural language, as writing Cypher queries might be challenging for non-technical users. In recent years, some studies have started working on the topic [5]; however, it needs a more thorough investigation.

Acknowledgement

This study was possible with the assistance of Dr. Malcolm Duncan, Medical Object Oriented Software Ltd, who shared the Diseases Database with us and answered our technical questions regarding the database. Also, this research has been partially funded by the NSERC Discovery Grant (RGPIN-2020-05869) in Canada and the CAISR Health (Knowledge Foundation grant 20200208 01H) in Sweden.

References

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

[2] G. Lagunes-Garcia, A. Rodriguez-Gonzalez, L. Prieto-Santamaria, E. P. Garcia del Valle, M. Zanin, and
E. Menasalvas-Ruiz. Disnet: a framework for extracting phenotypic disease information from public sources. PeerJ, 8(e8580), 2020.

[3] J. Pi˜nero, J. M. Ram´ırez-Anguita, J. Sa¨uch-Pitarch, F. Ronzano, E. Centeno, F. Sanz, and L. I. Furlong. The DisGeNET knowledge platform for disease genomics: 2019 update. Nucleic Acids Research, 48(D1):D845–D855, 11 2019.

[4] A. Saha, V. Pahuja, M. Khapra, K. Sankaranarayanan, and S. Chandar. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[5] C. Sun et al. A natural language interface for querying graph databases. PhD thesis, Massachusetts Institute of Technology, 2018.