Análise de Relação Doenças-Sintomas

Time Sugoi
Esdras Rodrigues do Carmo - RA: 170656
Gabriel Ryo Hioki - RA: 172434

Resumo

O problema estudado consiste na eficiência na busca em um conjunto de doenças e sintomas. Com uma busca eficiente, espera-se que a identificação de doenças a partir dos sintomas apresentados em um paciente seja mais precisa e veloz. Será utilizado análise de redes em um grafo de doenças e sintomas, com arestas relacionando doenças, sintomas e similaridades entre doenças. As doenças serão *clusterizadas* de acordo com o *score* de similaridade. Com isso, poderemos classificar as doenças e fazer uma busca mais assertiva no banco de dados. Os sintomas mais comuns serão ordenados utilizando um algoritmo de *PageRank*. Entretanto, outros métodos de buscas e armazenamento são utilizados além da análise de redes em grafo para poder-se comparar os benefícios e desvantagens de cada tipo. Os outros modelos analisados são o relacional, RDF e XML.

Requisitos do Modelo Conceitual

O modelo conceitual deve suportar o armazenamento de doenças e sintomas, assim como suas relações. Além disso, deve ter uma boa representação dos relacionamentos existentes, de modo a aumentar a eficiência da análise de dados.

É importante também manter o modelo simples e claro o bastante para que qualquer usuário consiga entendê-lo, mesmo que quando implementado seja utilizado outro modelo lógico mais otimizado, como por exemplo um banco de dados em grafos.

Fonte de Dados

O conjunto de dados é formado por 4 arquivos, sendo cada um deles uma tabela: doenças, sintomas, relacionamento entre doenças e sintomas, relacionamento de similaridade entre doenças.

Zhou, XueZhong et al. (2014). Human symptoms-disease network.

https://www.nature.com/articles/ncomms5212#s1

Modelagem Conceitual

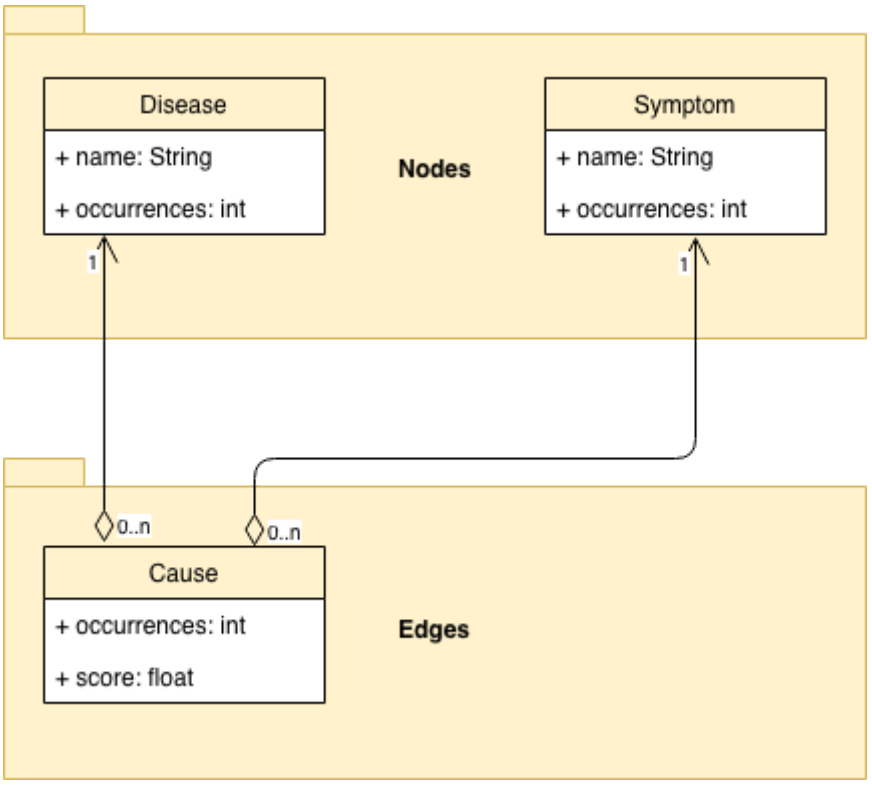

Figura 1: Modelo Conceitual em UML para XML

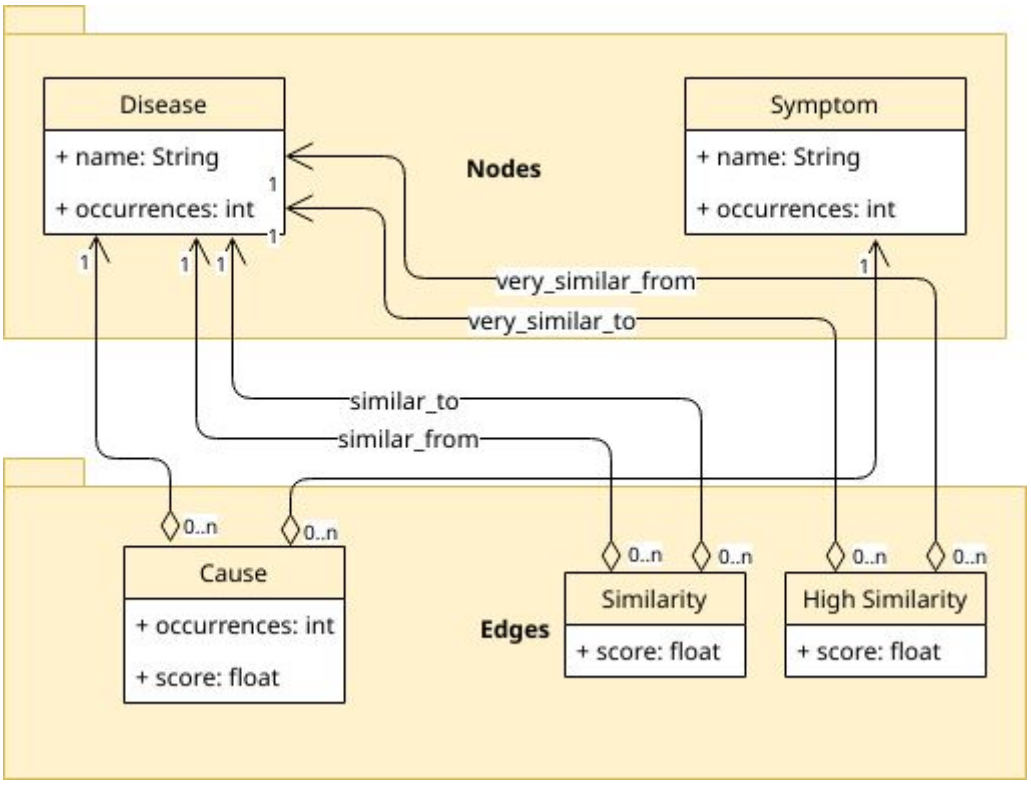

Figura 2: Modelo Conceitual em UML para RDF

Modelagem Lógica

Na análise do modelo XML, a relação doença-doença, que era dada como *Similarity*, não foi utilizada pelo motivo da ferramenta *Zorba*, utilizada para a análise neste tipo de modelo, não suportar a quantidade de dados da fonte que foi escolhida para este projeto. Assim, o modelo lógico fica:

- **Disease**(name, occurrences)
- **Symptom**(name, occurrences)
- **Cause**(disease, symptom, occurrences, score)
    - Chave Estrangeira: disease -> **Disease**
    - Chave Estrangeira: symptom -> **Symptom**

Na análise do modelo RDF, utilizamos a mesma modelagem lógica do banco de dados em grafo (Neo4J):

- **Disease**(name, occurrences)
- **Symptom**(name, occurrences)

- **Cause**(disease, symptom, occurrences, score)
    - Chave Estrangeira: disease -> **Disease**
    - Chave Estrangeira: symptom -> **Symptom**
- **Similarity**(disease_from, disease_to, score)
    - Chave Estrangeira: disease_from -> **Disease**
    - Chave Estrangeira: disease_to -> **Disease**
- **High Similarity**(very_similar_from, very_similar_to, score)
    - Chave Estrangeira: very_similar_from -> **Disease**
    - Chave Estrangeira: very_similar_to -> **Disease**

Explicação dos Benefícios do modelo RDF e XML

O modelo RDF é essencialmente a canonização de um grafo dirigido, assim tem todas as vantagens de estruturar informações usando grafos. Como este projeto era inicialmente voltado para análise em rede de grafos, este modelo facilita a análise sobre os dados escolhidos.

Em relação ao modelo XML, não há muita vantagem na utilização da mesma, uma vez que o conjunto de dados escolhidos possuem muitas instâncias de objeto, mas cada objeto possui poucos atributos. Dessa maneira, a vantagem da boa visualização do modelo XML não foi bem aproveitada e apenas buscas básicas e simples foram possíveis de ser realizadas.