Digital Healthcare Empowering Europeans
R. Cornet et al. (Eds.)

doi:10.3233/978-1-61499-512-8-975

Exploiting Semantic Predications in a Graph Database

Andrej KASTRIN[a,1], Thomas C. RINDFLESCH[b], Dejan DINEVSKI[c] and Dimitar HRISTOVSKI[d]
[a]*Faculty of Information Studies, Novo mesto, Slovenia*
[b]*National Library of Medicine, Bethesda, MD, USA*
[c]*Faculty of Medicine, University of Maribor, Maribor, Slovenia*
[d]*Faculty of Medicine, University of Ljubljana, Ljubljana, Slovenia*

Keywords. Databases, Data Mining, Semantics, Literature Based Discovery

Knowledge extraction using semantic relations is crucial for accurate and valid knowledge management in biomedicine. The Semantic MEDLINE Database (SemMedDB) contains semantic predications extracted with the SemRep semantic interpreter. Predications are structured as subject-predicate-object triples and can be represented as a directed network. Arguments correspond to UMLS Metathesaurus concepts, while predicates correspond to relations in the UMLS Semantic Network (e.g., SemRep extracts the predication `Ethionine-CAUSES-Lesion` from the sentence "Ethionine can trigger lesions."). SemMedDB has predications from all of MEDLINE and is available as a MySQL database. MySQL is generally efficient, but modeling networks using a relational database causes a large number of many-to-many relations. Complex join queries are then needed to retrieve such data. In this poster we present the Neo4j graph database as an alternative for storing, retrieving, and mining SemMedDB.

Neo4j is particularly useful for storing data structured as a network. Data objects are stored in the form of either a node or an edge. Neo4j is exceptionally scalable (i.e., several billion nodes on a single machine) and has an integrated pattern-matching-based query language (Cypher).

We import a large subset of SemMedDB into a Neo4j database using the standalone batch importer. Currently there are 247,582 unique concepts and 13,211,179 distinct relationships between them in the database. In the poster we systematically present different scenarios for importing, retrieving, and mining SemMedDB. An example scenario with Neo4j is the retrieval of all concepts related to Raynaud disease: "`MATCH (node1 {name:"Raynaud Disease"}) -[rel]-> (node2) RETURN node1, node2;`".

In future work we will systematically compare Neo4j with MySQL in terms of objective benchmark measures, including processing speed based on a predefined set of queries, disk space requirements, and scalability. We are currently working on a visualization module, which will greatly improve the user experience when mining SemMedDB. Our long-term interest is to employ Neo4j as a backend for literature-based discovery.

[1] Corresponding Author.