Towards Implementing Semantic Literature-Based Discovery with a Graph Database

E-mail: dimitar.hristovski@gmail.com

Dimitar Hristovski[1] , Andrej Kastrin[2], Dejan Dinevski[3], Thomas C. Rindesch[4]

[1]Faculty of Medicine, Ljubljana, Slovenia ,

[2]Faculty of Information Studies, Novo mesto, Slovenia;

[3]Faculty of Medicine, Maribor, Slovenia;

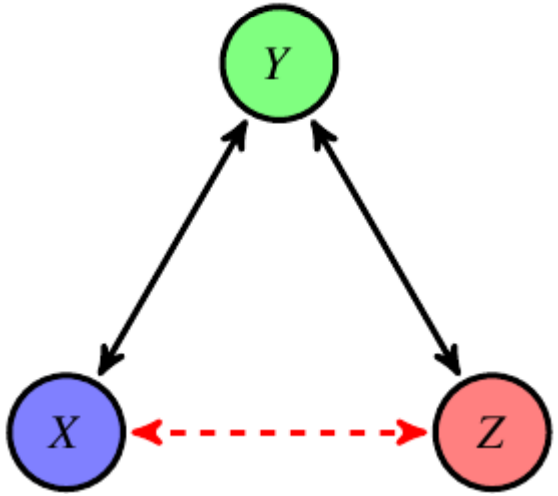
[4]National Library of Medicine, Bethesda, USA;

Text Mining

- Information extraction: Extract structured information from unstructured documents.
- Document summarization: Reduce documents to create a summary with most important parts.
- Question-Answering: Automatically answer questions posed by humans.
- Literature-based discovery

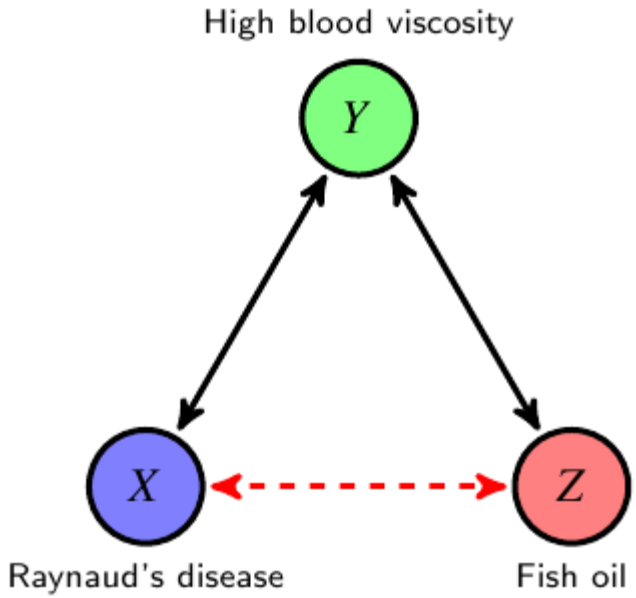
Literature-based Discovery (LBD)

- Methodology for generating hypotheses by uncovering implicit relationships from existing knowledge



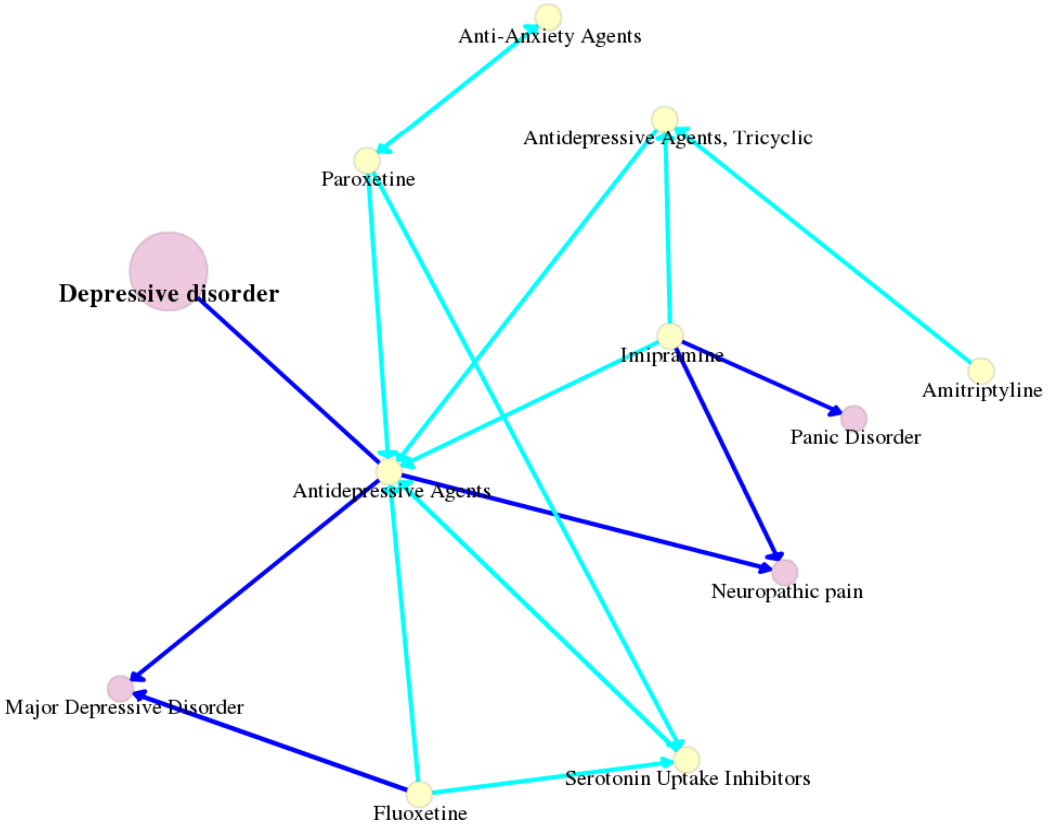
Swanson's LBD

- Raynaud's disease is associated with high blood viscosity
- Fish oil has been shown to lead to reduction in blood viscosity

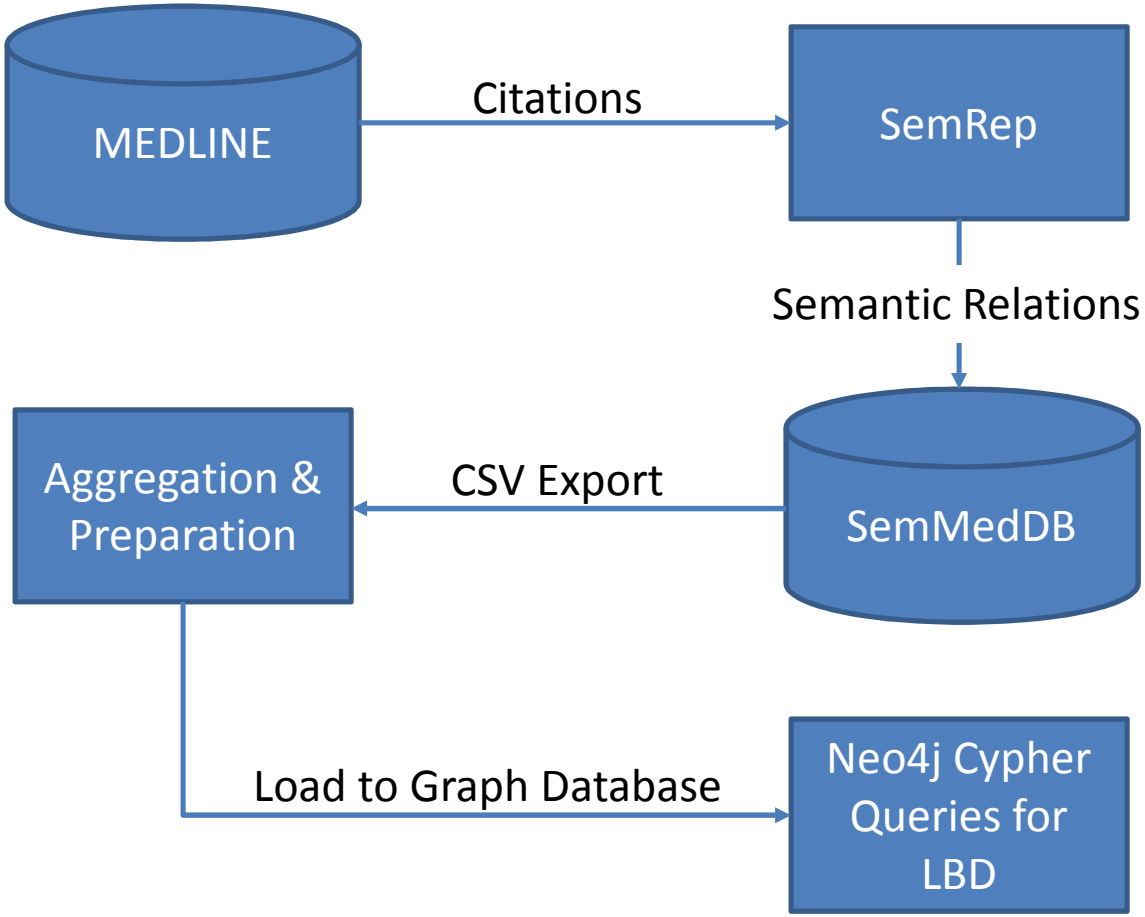


Representing Biomedical Knowledge as a Concept Graph

- Nodes: biomedical concepts
- Edges and/or arcs: relations between the concepts
- Concept relations:
  - Co-occurrences
  - semantic relations



From Documents to Concept Graph



Extracting Semantic Relations with SemRep

- SemRep is a natural language processing system that extracts semantic propositions from the biomedical research literature
- Example: From “dexamethasone is a potent inducer of multidrug resistance-associated protein expression in rat hepatocytes“ SemRep extracts:
  - Dexamethasone STIMULATES Multidrug Resistence-Associated Proteins
  - Multidrug Resistance-Associated Proteins PART_OF Rats
  - Hepatocytes PART_OF Rats

- SemMedDB - a mySQL database of extracted semantic relations from MEDLINE

Neo4j

- A native graph database
- Supports graph property data model
- Has declarative query language Cypher - uses ASCII-Art to represent graph patterns

From: http://dx.doi.org/10.1186/1742-4682-4-50

Export from SemMedDB

- 52 616 158 semantic relation instances exported
- CSV format

Aggregation and Loading with LOAD CSV

```
LOAD CSV FROM ’semmed_sub_rel_obj.txt’ AS line
WITH line
MERGE (c1:Concept {cui: line[0]})
        ON CREATE SET c1.name=line[1],
                c1.type=line[2], c1.freq=1
        ON MATCH SET c1.freq = c1.freq + 1
MERGE (c2:Concept {cui: line[4]})
        ON CREATE SET c2.name=line[5],
                c2.type=line[6], c2.freq=1
        ON MATCH SET c2.freq = c2.freq + 1
MERGE (c1)-[r:Relation {type:line[3]}]->(c2)
        ON CREATE SET r.freq = 1
        ON MATCH SET r.freq = r.freq + 1;
```

Aggregation and Loading with Import Tool

- Aggregation with AWK scripts
- Preparation of import files with AWK scripts and shell utilities (e.g. join, sort, ...)
- Stand alone batch import tool jexp (https://github.com/jexp/batch-import)
- Import worked very fast

Results – Graph Database Size

- 269 047 nodes (unique concepts)
- 14 150 952 relationships between the nodes (aggregated from 52 616 158 relation instances)
- 58 relationship types (e.g. TREATS, CAUSES, ...)
- 132 node labels used for semantic types

Implementing LBD with Cypher

- Most general LBD
- Finding novel treatments
- Generic “inhibit the cause of the disease” discovery pattern
- More specific version of “inhibit the cause of the disease”

Most General LBD

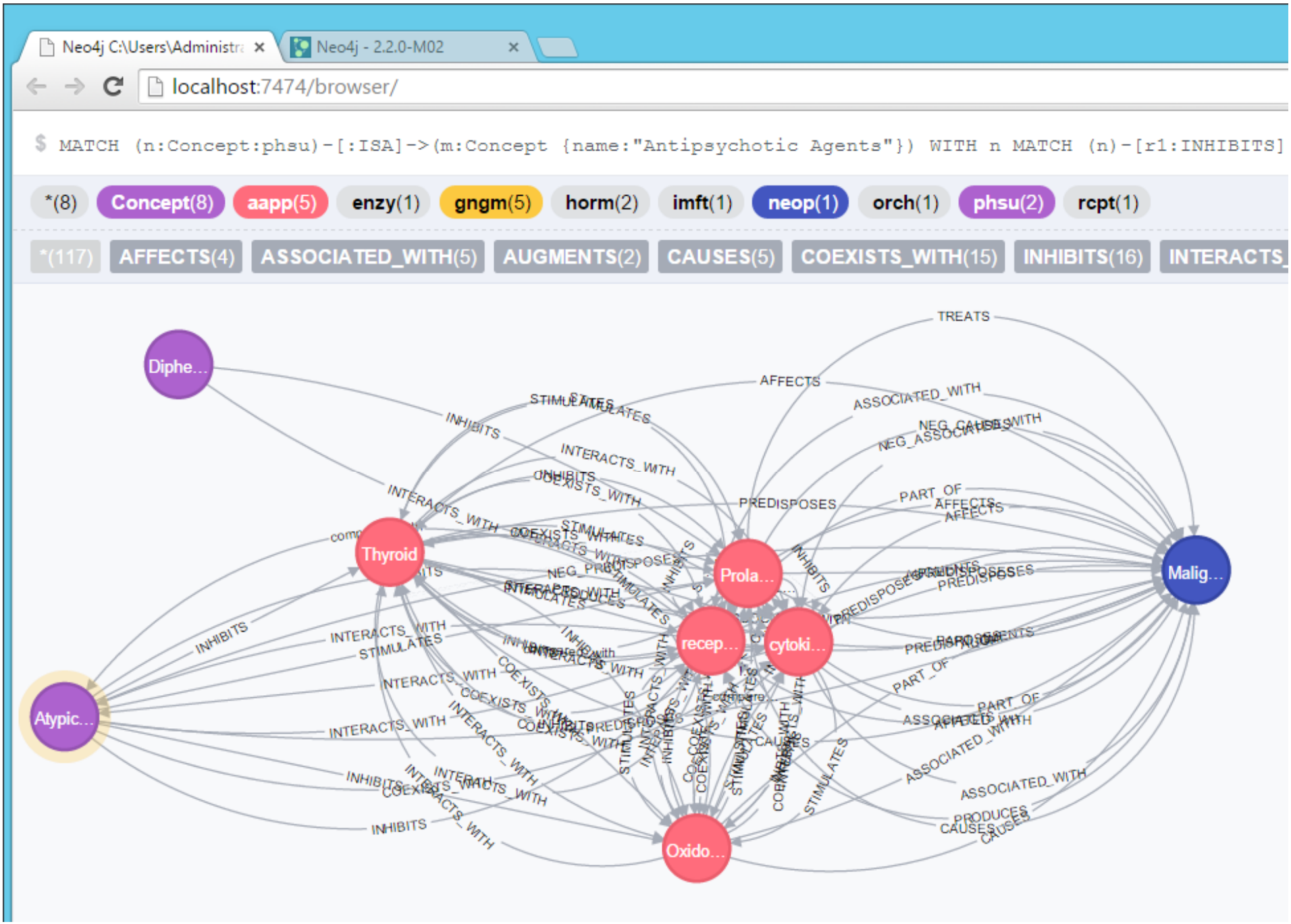
```
MATCH (x:Concept)--(y:Concept)--(z:Concept)
WHERE NOT (x)--(z)
RETURN x, y, z;
```

General Query for Finding Novel Treatments

```
MATCH (drug:Concept:phsu)-[r1]->(y)
        -[r2]->(disease:Concept:dsyn)
WHERE NOT (drug)-[:TREATS]->(disease)
RETURN drug, disease, count(y) AS y_count
DESC;
```

“Inhibit the Cause of the Disease” Discovery Pattern

```
MATCH (drug:phsu)-[:INHIBITS]->
        (gene:gngm)-[:CAUSES]-> (disease:dsyn)
WHERE NOT (drug)-[:TREATS]->(disease)
RETURN drug, gene, disease;
```


Neo4j C:\Users\Administra
Neo4j - 2.2.0-M02
localhost:7474/browser/
$ MATCH (n:Concept:phsu)-[:ISA]->(m:Concept {name:"Antipsychotic Agents"}) WITH n MATCH (n)-[r1:INHIBITS]
*(8)
Concept(8)
aapp(5)
enzy(1)
gngm(5)
horm(2)
imft(1)
neop(1)
orch(1)
phsu(2)
rcpt(1)
*(117)
AFFECTS(4)
ASSOCIATED_WITH(5)
AUGMENTS(2)
CAUSES(5)
COEXISTS_WITH(15)
INHIBITS(16)
INTERACTS_
Diphe...
Thyroid
Prola...
recep...
cytoki...
Malig...
Atypic...
Oxido...
TREATS
AFFECTS
ASSOCIATED_WITH
INHIBITS
STIMULATES
INTERACTS_WITH
COEXISTS_WITH
PREDISPOSES
PART_OF
PRODUCES
CAUSES

Discussion

- Challenges when loading into Neo4j
- Indexing confusion in Neo4j
- Fast performance with a small number of starting nodes
- Unpredictable performance with large number of starting nodes or when aggregation required

Future Work

- Performance evaluation and comparison: speed and storage
- Compare with: relational database(s) (e.g. mySQL), triple store (e.g. Virtuoso)
- Develop web application

Conclusions

- Graph database Neo4j suitable for representing biomedical knowledge needed for semantic LBD
- Query language Cypher is (relatively) easy to express LBD discovery patterns

More Specific Version of “Inhibit the Cause of the Disease”

```
MATCH (drug:Concept:phsu)-[:ISA]->
        (m:Concept {name:"Antipsychotic Agents"})
WITH drug
MATCH (drug)-[:INHIBITS]->
        (gene:gngm)-[:CAUSES]->(s:neop)
WHERE NOT (drug)-[:TREATS]->(s)
RETURN drug, count(distinct gene), count(distinct s);
```