Experimenting with Neural Networks and Graph Databases to Assist in Semantic Alignment of Clinical Outcomes Data of Stem Cell Transplantation within the NMDP

2019/04/23

Robinette Renner & Jane Pollack
Agenda

• Background
• Using Artificial Neural Networks
• Using Graph Databases
• Wrap up and Next Steps
The CIBMTR® (Center for International Blood and Marrow Transplant Research®) is a research collaboration between the National Marrow Donor Program® (NMDP)/Be The Match® and the Medical College of Wisconsin (MCW).
<table>
<thead>
<tr>
<th><strong>NMDP</strong></th>
<th><strong>CIBMTR</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Total number of unrelated transplants: 86,000</td>
<td>• Total number of Transplant/Stem Cell Patients: 500,000</td>
</tr>
<tr>
<td>• 6,100 in 2017 alone</td>
<td>• 400+ US Centers all contributing data</td>
</tr>
<tr>
<td>• 19 Million possible adult donors</td>
<td>• 1,000+ Publications related to Transplantation and Stem Cell Therapies</td>
</tr>
<tr>
<td>• 249,000 cord blood units (CBU)</td>
<td>• 27,000+ Units of Clinical Outcomes information (data points) over 50 years</td>
</tr>
<tr>
<td>• Collaborations across 46 countries</td>
<td></td>
</tr>
<tr>
<td>• 30 Million unrelated donors</td>
<td></td>
</tr>
<tr>
<td>• 742,000 CBU</td>
<td></td>
</tr>
</tbody>
</table>
Problem Statement

- Manually mapping to BRIDG is labor-intensive
- A team of 6 mapped 1,284 CDEs in 1 year
- We have more than 2,000 CDEs left to map
- Routine maintenance required to reflect changes to model and clinical practice
Data Management Requirements

• A good problem to have over 50 years
  – Improved Understanding of Blood Cancer
  – Long-Term Survival has increased
    • Long-Term follow-ups (10+ years)

• The consequence: Increased complexity of Data Collection and Management
  – Semantic experts are hard to grow
  – Pressure to focus on value-added tasks
Extensions

– GUI for BRIDG Mapping Facilitator Prototype
  • Experiment to extend FormsNet Metadata Mapper
  • Mask the Neural Network calls and return just the list of best guesses
  • Mask the Cypher Queries and return just the list of shortest paths

– Treat integration of BRIDG and HL7 FHIR as a Graph Problem
  • ClinFHIR already treats HL7 Bundles as graphs

– Treat integration of Ontologies as a Neural Network problem
  • Extend the work of Huang, et.al. to include semantics

– Treat mapping of ontologies to our domain of interest as a graph problem
  • So we can use graph-traversal algorithms to leverage semantics and map BRIDG-to-FHIR
CDE to Mapping Path (by Hand)

CDE:2682630
Name: Acute Myeloid Leukemia
Classification Type

1. Determine the Core Semantic
2. Determine path from Subject to Core Semantic
3. Determine Supporting Information
Problem Statement
CDE to BRIDG Mapping Path

Start with a CDE
• Start with a Common Data Element (CDE)
  • Defined by Metadata Analyst
  • ~Question on a Form

Determine the Core Semantic
• Determine the ‘end point’ of a path
  • Use Neural Networks
  • Select the most appropriate BRIDG Class

Determine how this relates to a Subject
• Traverse from Class to Class within the BRIDG Model
  • Exploit Graph Traversal Algorithms
  • Conform to BRIDG Grammatical Conventions

Add in Supportive Information
• Annotate with Subscripts, Attributes, Literals, and Subordinate Clauses
  • Enhance using Graph Similarity and Link Prediction Algorithms

CDE: 2682630
Name: Acute Myeloid Leukemia Classification Type

Class UML-Based Comprehensive BRIDG Model Diagram

Study Conduct Sub-Domain: Performance Diagnosis
- Study Conduct Sub-Domain: Performance Diagnosis
- bodySystemCode: CD [0-1]
- clinicalStageDescriptorCode: CD [1-1]
- diseaseStageCode: CD [1-1]
- diseaseStageCodingReasonCode: CD [1-1]
- etiologyStageDescriptorCode: CD [1-1]
- presentationStageDescriptorCode: CD [1-1]
- priorityAdmissionCode: CD [1-1]
- priorityAdmissionIndicator: BI [0-1]
Artificial Neural Networks
Defining the Core Semantic
CDE to BRIDG Mapping Path

- **Start with a CDE**
  - Start with a Common Data Element (CDE)
  - Defined by Metadata Analyst
  - Question on a Form

- **Determine the Core Semantic**
  - Determine the ‘end point’ of a path
  - Use Neural Networks
  - Select the most appropriate BRIDG Class

- **Traverse from Class to Class within the BRIDG Model**
  - Use Neural Networks
  - Select the most appropriate BRIDG Class

- **Add in Supportive Information**
  - Annotate with Subscripts, Attributes, Literals, and Subordinate Clauses
  - Enhance using Graph Similarity and Link Prediction Algorithms

---

**CDE: 2682630**
Name: Acute Myeloid Leukemia Classification Type

**Artificial Neural Networks**

---

CIBMTR®
CENTER FOR INTERNATIONAL BLOOD & MARROW TRANSPLANT RESEARCH
Example Mapping

<table>
<thead>
<tr>
<th>CDE</th>
<th>BRIDG Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acute Myeloid Leukemia Classification Type</td>
<td>PerformedDiagnosis</td>
</tr>
<tr>
<td>Chronic Myeloid Leukemia Classification Type</td>
<td>PerformedDiagnosis</td>
</tr>
</tbody>
</table>
Solution

Artificial Neural Networks

• Semi-automated mapping
• Use artificial neural networks to predict possible BRIDG classes
• Artificial neural networks
  – intended to replicate the way humans learn
  – excellent tool for finding patterns which are too complex for a human programmer to extract
## Accuracy

<table>
<thead>
<tr>
<th>Number of BRIDG Classes Returned</th>
<th>Training set: Verification set 3:1</th>
<th>Training set: Verification set 9:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.99%</td>
<td>41.52%</td>
</tr>
<tr>
<td>2</td>
<td>51.96%</td>
<td>63.16%</td>
</tr>
<tr>
<td>3</td>
<td>64.71%</td>
<td>73.10%</td>
</tr>
<tr>
<td>4</td>
<td>71.90%</td>
<td>80.10%</td>
</tr>
<tr>
<td>5</td>
<td>76.14%</td>
<td>83.04%</td>
</tr>
<tr>
<td>6</td>
<td>82.03%</td>
<td>85.38%</td>
</tr>
<tr>
<td>7</td>
<td>85.29%</td>
<td>87.72%</td>
</tr>
<tr>
<td>8</td>
<td>86.93%</td>
<td>90.06%</td>
</tr>
<tr>
<td>9</td>
<td>89.22%</td>
<td>91.23%</td>
</tr>
<tr>
<td>10</td>
<td>92.16%</td>
<td>94.15%</td>
</tr>
</tbody>
</table>
Limitations of Algorithm

• Pattern recognition only, no semantics
• Only maps to BRIDG classes with sufficient training data
Graph Databases & Algorithms
CDE to BRIDG Mapping Path

- **Start with a Common Data Element (CDE)**
  - Defined by Metadata Analyst
  - ~Question on a Form

- **Determine the Core Semantic**
  - Determine the ‘end point’ of a path
  - Use Neural Networks
  - Select the most appropriate BRIDG Class

- **Determine how this relates to a Subject**
  - Traverse from Class to Class within the BRIDG Model
  - Exploit Graph Traversal Algorithms
  - Conform to BRIDG Grammatical Conventions

- **Add in Supportive Information**
  - Annotate with Subscripts, Attributes, Literals, and Subordinate Clauses
  - Enhance using Graph Similarity and Link Prediction Algorithms

---

**Graph Databases & Traversal Algorithms**

- CDE:2682630
  - Name: Acute Myeloid Leukemia
  - Classification Type

- **Graph Databases & Traversal Algorithms**
  - Graph Databases &  
    Traversal Algorithms
What is a Graph Database?

• **graph theory** is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices, nodes, or points which are connected by edges, arcs, or lines.

• A Graph Database is an implementation of modeling data in a graph format.

• Neo4j is a popular open-source graph-database platform

https://tdwi.org/articles/2017/03/14/good-bad-and-hype-about-graph-databases-for-mdm.aspx
Step 1: Extract BRIDG 5.2 Entities, Attributes, Relationships into CSV files
Treating BRIDG (a UML model) like a Graph

This is an Improvement?
Using the BRIDG Model to build an Instance Graph

Mapping Path for CDE 2527897
Subject > BiologicEntity > BiologicEntityIdentifier.identifier WHERE BiologicEntityIdentifier.typeCode = “CRID”
How does a CDE Mapping Path get converted to a Graph?

Components for CDE 2527897

<table>
<thead>
<tr>
<th>orig_component_text</th>
<th>valid_component_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>ENTITY</td>
</tr>
<tr>
<td>&gt;</td>
<td>ASSOCIATION</td>
</tr>
<tr>
<td>BiologicEntity</td>
<td>ENTITY</td>
</tr>
<tr>
<td>&gt;</td>
<td>ASSOCIATION</td>
</tr>
<tr>
<td>BiologicEntityIdentifier</td>
<td>ENTITY</td>
</tr>
<tr>
<td>.</td>
<td>ENTITY.ATTRIBUTE</td>
</tr>
<tr>
<td>identifier</td>
<td>DESIGNATION</td>
</tr>
<tr>
<td>WHERE</td>
<td>FILTER KEYWORD</td>
</tr>
<tr>
<td>BiologicEntityIdentifier</td>
<td>ENTITY</td>
</tr>
<tr>
<td>.</td>
<td>ENTITY.ATTRIBUTE</td>
</tr>
<tr>
<td>typeCode</td>
<td>DESIGNATION</td>
</tr>
<tr>
<td>=</td>
<td>OPERATOR</td>
</tr>
<tr>
<td>&quot;CRID&quot;</td>
<td>LITERAL</td>
</tr>
</tbody>
</table>
Problem B: Defining the Mapping Path

Graphical Representation of the shortest paths between ‘Subject’ and ‘PerformedSubstanceAdministration’

**Shortest Paths** between ‘Subject’ and ‘PerformedSubstanceAdministration’

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Subject, Activity, PerformedActivity, PerformedSubstanceAdministration]</td>
<td>['be participated in by', 'specializes', 'have start evaluated in relation to']</td>
</tr>
<tr>
<td>[Subject, Activity, PerformedActivity, PerformedSubstanceAdministration]</td>
<td>['be participated in by', 'specializes', 'have end evaluated in relation to']</td>
</tr>
</tbody>
</table>

Cypher Query:
```
MATCH p=allShortestPaths((c1:ClassName {name:'Subject', version:'5.2'})-[:ASSOCIATION_WITH|DESCENDENT_OF*..10]-(c2:ClassName {name:'PerformedSubstanceAdministration',version:'5.2'}))
WHERE NONE(x IN NODES(p) WHERE x:ClassName AND x.name = 'Product')
RETURN extract(n in nodes(p) | n.name) as e1, extract(p1 in relationships(p) | p1.relationship) as e2, nodes(p), relationships(p)
```
C.1 Validate BRIDG Mapping Paths: Validate Entities

- Mapping Path: `PerformedDiagnosis` > `PerformedSubstanceAdministration` [1] > `DefinedSubstanceAdministration.nameCode` WHERE `PerformedDiagnosis.value = "Chronic Myelogenous Leukemia(CML)"` AND `PerformedSubstanceAdministration.dateRange [1] IS LESS THAN {transplant date}

- Cypher Query: MATCH (c:ClassName {name: "PerformedDiagnosis", version: "3..."}) RETURN count(*)

Given the conventions of a BRIDG Mapping Path, make sure all Classes, Attributes, and Relationships are correct and follow correct UML Inheritance structures

Graphical Representation of Entity Validations
C.2: Validate Mapping Paths: Validate Attributes for an Entity

- **Mapping Path:** PerformedDiagnosis > PerformedSubstanceAdministration [1] > DefinedSubstanceAdministration.nameCode
  WHERE PerformedDiagnosis.value = "Chronic Myelogenous Leukemia(CML)" AND PerformedSubstanceAdministration.dateRange [1] IS LESS THAN {transplant date}

- Cypher Query: MATCH (c:ClassName { name:"PerformedSubstanceAdministration", version:"3.2"}) - [:DESCENDENT_OF* { version:"3.2"}] -> (c1:ClassName { version:"3.2"}) - [:HAS {version:"3.2"}] - (a:Attribute { name:"dateRange", version:"3.2"}) return *

Given the conventions of a BRIDG Mapping Path, make sure all Classes, Attributes, and Relationships are correct and follow correct UML Inheritance structures.

**Graphical Representation of Attribute Validation**
Step C.3: Validate Mapping Path: Validate Entity Associations

- **Mapping Path:**
  
  ```
 PerformedDiagnosis > PerformedSubstanceAdministration [1] > DefinedSubstanceAdministration
  ```

  Where
  ```
  PerformedDiagnosis.value = "Chronic Myelogenous Leukemia(CML)" AND PerformedSubstanceAdministration.dat eRange [1] IS LESS THAN {transplant date}
  ```

  - **Cypher Query:**
    ```
    MATCH (c1a:ClassName {name:"PerformedSubstanceAdministration", version:"3.2"}) - [:DESCENDENT_OF* { version:"3.2"}] - (c1:ClassName { version:"3.2"})
    WITH c1 AS x MATCH (x) - [:ASSOCIATION_WITH { version:"3.2"}] - (c2:ClassName { version:"3.2"})
    MATCH (x1) - [:DESCENDENT_OF* { version:"3.2"}] - (c2a:ClassName { name:"DefinedSubstanceAdministration", version:"3.2"})
    RETURN "BothDescendent AS A1, x1.name AS LeftEntity, c2a.name AS RightEntity"
    ```

  Given the conventions of a BRIDG Mapping Path, make sure all Classes, Attributes, and Relationships are correct and follow correct UML Inheritance structures.

  Graphical Representation of Entity Associations:
Step 4: Updating a Mapping Path

- Given multiple versions of BRIDG Models, convert no-longer-valid (or no-longer-desired) paths to another

<table>
<thead>
<tr>
<th>Version</th>
<th>Mapping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td><code>PerformedObservationResult.value (ANY=&gt;CD) [eg &quot;frozen&quot;] WHERE PerformedObservation &gt; DefinedObservation.nameCode = &quot;Evaluate Container Condition&quot; AND PerformedObservation &gt; Subject [Package] [container]</code></td>
</tr>
<tr>
<td>5.2</td>
<td><code>PerformedObservationResult.value (ANY=&gt;CD) [eg &quot;frozen&quot;] WHERE PerformedObservation &gt; DefinedObservation.nameCode = &quot;Evaluate Container Condition&quot; AND PerformedObservation &gt; Subject [Package] [container]</code></td>
</tr>
</tbody>
</table>
CDE to BRIDG Mapping Path

Start with a Common Data Element (CDE)
- Defined by Metadata Analyst
- ~Question on a Form

Determine the Core Semantic
- Determine the ‘end point’ of a path
- Use Neural Networks
- Select the most appropriate BRIDG Class

Traverse from Class to Class within the BRIDG Model
- Exploit Graph Traversal Algorithms
- Conform to BRIDG Grammatical Conventions

Annotate with Subscripts, Attributes, Literals, and Subordinate Clauses
- Enhance using Graph Similarity and Link Prediction Algorithms

Graph Databases & Link Similarity Algorithms
Next Steps: Metadata-Driven Architecture

Use Graph-Theory Link Prediction algorithms to help combine paths together:
Link Common Sub-Graphs → HL7 FHIR Resource Equivalents

Cypher Query:
MATCH (n:PathQualifier {version:'5.2'})
WHERE ((upper(n.value) contains 'AML')
OR (upper(n.value) contains 'ACUTE MYELOGENOUS LEUKEMIA'))
MATCH (n1 {version: '5.2'}) WHERE
NOT ("ClassName" in labels(n1))
OR "Attribute" in labels(n1)
OR "Component" in labels(n1))
WITH n, n1
WHERE n1.pathSourceIID = n.pathSourceIID return n1 LIMIT 500
Facilitated Mapping in NMDP/CIBMTR
Prototype based on FN Mapping Tool

GUI for BRIDG Mapping Facilitator Prototype

Experiment to extend FormsNet Metadata Mapper

• Mask the Neural Network calls and return just the list of best guesses
• Mask the Cypher Queries and return just the list of shortest paths
Prototype based on FN Mapping Tool
Phase 2

Populating BRIDG Class:

- Access Neural Network to get top ten classes
- Access Graph Database to get definitions of classes
- Metadata Analyst selects best choice of BRIDG Class

MATCH (c:ClassName {name:"PerformedObservationResult", version:"5.2"})-[:DESCENDENT_OF]-() RETURN c1.name AS name, c1.notes AS notes
UNION MATCH (c2:ClassName {name:"PerformedObservationResult", version:"5.2"}) RETURN c2.name AS name, c2.notes AS notes
Prototype based on FN Mapping Tool
Phase 2

Populating BRIDG Attribute:

• Access Graph Database to get attributes and their definitions
• Metadata Analyst selects best choice of BRIDG attribute

MATCH (c:ClassName{name:"PerformedObservationResult", version:"5.2"}){[:HAS]-(c1:Attribute)} RETURN c1.name AS name, c1.notes AS notes

name | notes
--- | ---
identifier | DEFINITION: The unique symbol that establishes identity of the observation
confidenceCode | DEFINITION: A coded value specifying the degree of privacy applicable for the observation
comment | DEFINITION: Additional description of the observation result. EXAMPLE(S): Other information about the result
value | DEFINITION: Data or information that is determined by an act of observation
valueCodeModifiedText | DEFINITION: A character string that is a revision of the original text observation
baselineIndicator | DEFINITION: Specifies whether the result is a starting point to which other results are compared
uncertaintyCode | DEFINITION: A coded value specifying whether and to what degree this evaluation is uncertain
reportedDate | DEFINITION: The date (and time) on which the result is reported. EXAMPLE(S): Date of issuance
valueNullFlavorReason | DEFINITION: The text and/or code that describes why no result was provided
createdDate | DEFINITION: The date (and time) on which the result is created. EXAMPLE(S): Date of issuance
typeCode | DEFINITION: A coded value specifying the kind of observation result. EXAMPLE(S): Type of evaluation
Prototype based on FN Mapping Tool

Phase 1

Determining Best Path from Subject to the Core Semantic:

- Access Graph Database to call allShortestPaths
- Metadata Analyst selects best choice of BRIDG path

MATCH (c:ClassName)
{name:"PerformedObservationResult", version:"5.2"})-[HAS]-(c1:Attribute) RETURN c1.name AS name, c1.notes AS notes

- "Subject", "Activity", "PerformedActivity", "PerformedObservationResult"
- "Subject", "Activity", "PerformedActivity", "PerformedObservationResult"
- "Subject", "DocumentAuthor", "DocumentVersion", "PerformedObservationResult"
- "Subject", "Organization", "PointOfContact", "PerformedObservationResult"
- "Subject", "Organization", "PointOfContact", "PerformedObservationResult"
Prototype based on FN Mapping Tool
Phase 2

Creating SubClauses:

• Access Graph Database to call allShortestPaths from CoreSemantic to other classes

• Metadata Analyst selects best choice of BRIDG path

• Metadata Analyst inputs literals and subscripts to complete path

MATCH (c:ClassName {name:"PerformedObservationResult", version:"5.2"})-[HAS]->(c1:Attribute) RETURN c1.name AS name, c1.notes AS notes
Questions