Knowledge Graph Evaluation Service

Amar Viswanathan
Sabbir Rashid
Ian Gross

kannaarpi.edu
rashis2rpi.edu
grossi2rpi.edu

Ontology Engineering 2017
Tetherless World Constellation
Rensselaer Polytechnic Institute

April 24, 2017
Acknowledgements

- **Deborah McGuinness** - Senior Constellation Chair, TWC & Professor of Computer and Cognitive Science
- **Elisa Kendall** - Member, OMG Architecture Board & co-chair, Ontology Definition Metamodel (ODM) RTF at the Object Management Group
- **Jim McCusker** - Director of Data Operations, TWC
- **Mihai Surdeanu** - Professor of Computer Science, Computational Language Understanding Lab, University of Arizona
- **Tom Hicks** - Senior Research Programmer, Computational Language Understanding Lab, University of Arizona
Motivation

- IE toolkits can extract semi-structured (XML, JSON) Information which may not be reliable
  - Source Validation
  - Incorrect Answers to Obvious Questions
  - Extraction Errors
  - Human Intervention
- IE toolkit ODIN [Valenzuela-Escárcega et al., 2015]
  - Extracts information from PubMed Documents
    - Entities
    - Events
    - Contexts
    - Sentences/Passages
- Evaluate Results of IE Generated Knowledge Graph to Test for Inconsistencies
  - Logical Inaccuracies
  - Semantic Type Mismatch
Related Work I

- **Evaluation Techniques Survey** [Brank et al., 2005]
  - Ontology evaluation in terms of categories
    - *Golden Standard* of Evaluation based on Comparison to another Reference Ontology
    - Comparison with Domain Data
    - Human Assessment
  - Level/layer based Evaluation Approach
    - Vocabulary
    - Hierarchy Structure
    - Syntax or Semantic Relations
    - External Ontology References

- **Chimaera** [McGuinness et al., 2000]
  - Web-based User Interface
  - Automated Merging of Multiple Ontologies
  - Ontology Diagnosis for Inconsistencies
  - Testing
Related Work II

- Incompleteness and Taxonomic Analysis
- Semantic and Syntactic Checks

- The OntOlogy Pitfall Scanner! (OOPS!) [Gómez-Péreza, 2009]
  - Targeted both Newcomers and Domain experts
  - Provided a list of Pitfalls in Ontology Creation and Management
  - Evaluation based on
    - Structural Dimensions
    - Functional Dimensions
    - Usability-Profiling Dimensions
    - Consistency
    - Completeness
    - Conciseness

- Instance Data Evaluation [Tao et al., 2009]
- Linked Data Quality Survey [Zaveri et al., 2016]
Use Case Description

Evaluating a Knowledge Extraction system is generally done through the lens of precision, recall and the extended F1 measure. However, with these systems aiming to create Knowledge Graphs with accurate information, it is worth exploring measures of correctness of such Graphs.

- **Title** - Knowledge Graph Evaluation System
- **Source** - ODIN IE Toolkit, UAZ
- **Goal** - Detect and Evaluate Inconsistencies or Potential Incorrect Labels in an IE Generated Knowledge Graph by using a supporting Ontology to identify Potential Errors
- **Scope** - IE Output on publications from PubMed and other Biomedical domain related conferences
- **Requirement** - The evaluation requires the inclusion of an Ontology that defines the schema for Creating and Querying KG triples from the Disparate Extractions, and contains Rules for Detecting Inconsistencies.
High Level Ontology Structure: Schema Design

Figure 1: Schema Diagram
High Level Ontology Structure: Inconsistency Hierarchy

Figure 2: Inconsistency Hierarchy
Information Extraction

Figure 3: Information Extraction
Knowledge Graph Generation

![Diagram of Knowledge Graph Generation]

- **XML**
  - Extracted Entities
  - Extracted Relations
  - Extracted Processes

- **Knowledge Graph Generator**
  - Inputs: Extracted Entities, Extracted Relations, Extracted Processes
  - Outputs: Knowledge Graph, RDF/XML (Supporting Ontology)

Figure 4: Knowledge Graph Generation
Knowledge Graph Evaluation System
Figure 6: System Diagram
Ontology Usage in Relation to Competency Questions

- Ontology used as a Schema to transform IE outputs to RDF
- Predicates allow for precise querying
- Inconsistency concepts define possible errors to check for
How many entity mentions are about CLTA-4? What sentences do they come from and what are their extracted mention types? What are equivalent classes to that individual what are the map types for those equivalent classes?
?s rdf:type kgcs:EntityMention .
?s kgcs:hasMentionType ?mtype .
?s kgcs:fromSentence ?sent .
?e owl:sameAs ?s .
FILTER (?slabel="CTLA-4").
# CTLA-4 ; ATP ; pH ; GDNF ; P200 ; p57 ; GP210 ; CD200
### Example Table Results: CTLA-4 Entity Mentions

<table>
<thead>
<tr>
<th>mention</th>
<th>mention label</th>
<th>From sentence</th>
<th>Mention type</th>
<th>Equivalent class</th>
<th>Equivalent class type</th>
</tr>
</thead>
</table>
Competency Question #2

How many event mentions are about ATP? What sentences do they come from and what are their extracted mention types? What are the extracted mention subtype? What are entity mentions that are also extracted from the given sentence?
(GROUP_CONCAT(distinct ?emlabel ; SEPARATOR="", ") as ?emlabels)
WHERE{
  ?s rdf:type kgcs:EventMention .
  ?s kgcs:hasMentionType ?mtype .
OPTIONAL { ?s kgcs:hasMentionSubType ?mstype .}
  ?s kgcs:fromSentence ?sent .
OPTIONAL {
  ?em rdfs:label ?emlabel . }
FILTER (?slabel="ATP").
# interferon-g; cholesterol ; DAF-12 ; ERK1 ; protein kinase C ; apoptosis } GROUP BY ?slabel ?mtype ?s ?e ?etype ?sent ?mstype ?emlabels
<table>
<thead>
<tr>
<th>Event mention</th>
<th>Event mention label</th>
<th>From sentence</th>
<th>Mention type</th>
<th>Mention subtype</th>
<th>Entity mention labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>KGCS:evem-PMC516774-U</td>
<td>ATP</td>
<td>KGCS:sent-PMC516774-U</td>
<td>activation</td>
<td>negative-activation</td>
<td>UO126 (25 muM), ATP</td>
</tr>
<tr>
<td>AZ-r1-516774-85-18</td>
<td></td>
<td>KGCS:sent-PMC516774-U</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KGCS:evem-PMC546331-U</td>
<td>ATP</td>
<td>KGCS:sent-PMC546331-U</td>
<td>protein-modification</td>
<td>hydrolysis</td>
<td>cat, RecA, ATP</td>
</tr>
<tr>
<td>AZ-r1-546331-156-54</td>
<td></td>
<td>KGCS:sent-PMC546331-U</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KGCS:evem-PMC546331-U</td>
<td>ATP</td>
<td>KGCS:sent-PMC546331-U</td>
<td>protein-modification</td>
<td>hydrolysis</td>
<td>cat, ATP</td>
</tr>
<tr>
<td>AZ-r1-546331-156-78</td>
<td></td>
<td>KGCS:sent-PMC546331-U</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KGCS:evem-PMC552968-U</td>
<td>ATP</td>
<td>KGCS:sent-PMC552968-U</td>
<td>complex-assembly</td>
<td>null</td>
<td>MSH2-MSH6, Cd, ATP</td>
</tr>
<tr>
<td>AZ-r1-552968-69-24</td>
<td></td>
<td>KGCS:sent-PMC552968-U</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What are the number of distinct mention types extracted for an entity mention?
SELECT ?label
(GROUP_CONCAT(distinct ?type ; SEPARATOR=", ") as ?types)
(COUNT(distinct ?type) as ?countType) WHERE {
?s a kgcs:EntityMention .
?s kgcs:hasMentionType ?type .
} GROUP BY ?label ?types ORDER BY DESC(?countType)
### Example Table Results: Event Mentions Distinct Types Count

<table>
<thead>
<tr>
<th>Event mention</th>
<th>Mention types</th>
<th>Mention subtypes</th>
<th>Type count</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAK2</td>
<td>protein-modification, activation, complex-assembly, regulation, controller</td>
<td>phosphorylation, negative-activation, null, positive-activation, positive-regulation, autophosphorylation</td>
<td>5</td>
</tr>
<tr>
<td>EGF</td>
<td>activation, controller, regulation, complex-assembly, protein-modification</td>
<td>positive-activation, positive-regulation, null, negative-activation, phosphorylation, negative-regulation</td>
<td>5</td>
</tr>
<tr>
<td>Tat</td>
<td>protein-modification, activation, complex-assembly, theme</td>
<td>deacetylation, phosphorylation, positive-activation, null, acetylation, negative-activation</td>
<td>4</td>
</tr>
<tr>
<td>JNK</td>
<td>protein-modification, regulation, complex-assembly, activation</td>
<td>phosphorylation, positive-regulation, null, positive-activation, negative-activation</td>
<td>4</td>
</tr>
</tbody>
</table>
Given a Knowledge Graph generated from PubMed documents, is the graph consistent based on the rules defined in the supporting ontology?

What conditions are required for a Misclassification of an Entity to be detected by the system?

Given the results of the Knowledge Graph Evaluation System, what would the inconsistent graph look like?

If two different sources define a single class in a contradictory fashion to one another, how do you know which source is more likely to be correct?

Which entity extractions are mapped to disjoint types?
Is the graph Consistent based on the rules defined in the Supporting Ontology?

If there are no violations of any instances to the basic rules, then the system would report a consistent graph.

- **Misclassification of Entity** - A drug is misclassified in terms of usage. For example, let's take the drug, Librium. Librium is ofDrugType Benzodiazepine, which is known to be a sedative type of drug. Librium also came back as ofDrugType Antihistamine because Antihistamines have a property that they can control anxiety (which is the purpose of Librium). However, Librium is not Antihistamine. This showcases where Librium classified as an individual from two disjoint concepts, leading to the misclassification result.
Which entity extractions are mapped to disjoint types?

- `ment-PMC524479-UAZ-r1-524479-55-73 rdfs:label P200 kgcs:hasMentionType Protein kgcs:fromSentence sent-PMC524479-UAZ-r1-524479-55`
- `ment-PMC524479-UAZ-r1-524479-55-74 rdfs:label P200 kgcs:hasMentionType Site kgcs:fromSentence sent-PMC524479-UAZ-r1-524479-55`

Two individuals extracted from the same sentence have different values for `kgcs:hasMentionType`.

Disjointness rule: `disjointWith(Protein,Site)` - Individual entity mentions will trigger corresponding disjointness inconsistencies.

Assumes that a protein cannot be a site, which may not necessarily be true.

Consider more obvious cases, such as `disjointWith(Protein, Bioprocess)`.
Representation Modeling: Mention Types

Figure 10: Mention Types
Representation Modeling: Event and Entity Mention Types

Entity Mention Types
- protein
- family
- simple-chemical
- bioprocess
- site
- cellular-component
- gene-or-gene-product
- generic-entity
- species
- cell-line

Event Mentions Types
- activation
- protein-modification
- complex-assembly
- controlled
- regulation
- theme
- translocation
- controller

Event Mention Subtypes
- positive-activation
- negative-activation
- phosphorylation
- positive-regulation
- dephosphorylation
- acetylation
- hydrolysis
- methylation
- translocation
- deacetylation
Figure 11: Example Entity Mention
Representation Modeling: Alternative Knowledge Methods

- **Mention Types**
  - The hierarchy could have been organized differently
  - Our current method of having more than 10 EntityMentionTypes and EventMentionSubTypes can be considered inefficient modeling
  - Example: Overarching types like Organism and Process

- **Entity Mentions**
  - We could have organized the data differently
  - All entities are instances of Entity types
  - Entity Types have a label which corresponds to the entity mention
Results from Oops! Evaluation

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical**: It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important**: Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor**: It is not really a problem, but by correcting it we will make the ontology nicer.

### Results for P36: URI contains file extension.

<table>
<thead>
<tr>
<th>ontology*</th>
<th>Minor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ontolog...</strong></td>
<td>()</td>
</tr>
</tbody>
</table>

This pitfall occurs if file extensions such as ".owl", ".rdf", ".ttl", ".n3" and ".rdfxml" are included in an ontology URI. This pitfall is related with the recommendations provided in [9].

*This pitfall applies to the ontology in general instead of specific elements.

### Results for P38: No OWL ontology declaration.

<table>
<thead>
<tr>
<th>ontology*</th>
<th>Important</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ontolog...</strong></td>
<td>()</td>
</tr>
</tbody>
</table>

This pitfall consists in not declaring the owl:Ontology tag, which provides the ontology metadata. The owl:Ontology tag aims at gathering metadata about a given ontology such as version information, license, provenance, creation date, and so on. It is also used to declare the inclusion of other ontologies.

*This pitfall applies to the ontology in general instead of specific elements.

### Results for P39: Ambiguous namespace.

<table>
<thead>
<tr>
<th>ontology*</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ontolog...</strong></td>
<td>()</td>
</tr>
</tbody>
</table>

This pitfall consists in declaring neither the ontology URI nor the xml:base namespace. If this is the case, the ontology namespace is matched to the file location. This situation is not desirable, as the location of a file might change while the ontology should remain stable, as proposed in [12].

*This pitfall applies to the ontology in general instead of specific elements.
Discussion of Oops! Results

Result Explanation

- The P36 Minor error is due to the fact that our published ontology on our website includes the .rdf extension. If we can publish without this extension, the error would be resolved.
- The P38 Important error is because our ontology is apparently missing an owl declaration. We should probably add that in there.
- The P39 Critical error is due to the fact that we included the prefix for xml:base in our Entity Declaration. This declaration can be removed in order to resolve the error.

These results were very reasonable. We made the change necessary to rectify the critical error.

Some changes we would like to see are more descriptive explanations and the list error filters should be more than simple key words.
Next Steps

- Updates to Inconsistency Hierarchy to include rules for each type of defined Inconsistency
- Add additional rules to detect inconsistencies in Event Mentions
- Leverage supporting Domain Ontologies to include Domain Specific Rules
- Consult with a Domain Expert to see which classes can be modeled as Disjoint
References I


Questions?