I. Use Case Description	
Use Case Name	Knowledge Graph Evaluation System
Use Case Identifier	OE2017-KGES-01.13
Source	Amar Viswanathan, Sabbir Rashid, Ian Gross
Point of Contact	Amar Viswanathan, kannaa@rpi.edu
Creation / Revision Date	Created 02/04/2017 / Revised 05/02/2017
Associated Documents	Software documentation, links to evaluation

II. Use Case Summary	
Goal	Information Extraction (IE) toolkits using Entity, Event and Relationship Extraction procedures, extract information in the form of entities, events and relationships, respectively. Since these are independent tasks, the outputs are also disparate documents which are evaluated individually for the precision and recall metrics. While precision and recall metrics calculate the statistical accuracy, they miss out on finding obvious semantic errors. If these outputs are converted to a RDF based Knowledge Graph, it would be possible to use the added semantics to look for said errors. Thus, the goal of this project is to detect and evaluate inconsistencies or potential incorrect labels in the resulting Knowledge Graph by using a supporting Ontology to identify potential errors.
Requirements	In order to meet the requirements of our goal, we must develop a framework for Knowledge Graph (KG) evaluation. This will include an Ontology that defines the schema for creating KG triples from the disparate extractions. While the extraction framework doesn't have to be created (an existing framework will be used, such as OLLIE or ODIN), the supporting ontology will have be engineered. The Ontology will be populated with terms from from the vocabulary of the individual extraction tasks, and it will also include terms that provide the schema necessary for integrating the outputs.
Scope	The scope of the system involves evaluating the correctness of a Knowledge Graph by leveraging supporting ontologies to check for inconsistencies. This may include a knowledge graph that was generated from an IE toolkit, manually by an ontologist or specialist, or a combination of a human and machine created graph, where an ontologist may have curated an automatically generated KG. The targeted audience for such a system would include those interested in evaluating the semantic capabilities of IE toolkit outputs. This may include end users, IE developers/evaluators or ontology engineers. As different IE tools have outputs in different formats, the initial scope of the system includes to pick up one tool (in this case, ODIN), for a particular domain (in this case, Bio-Med) and then combine all the outputs produced by them to a meaningful representation, i.e. map them to standard terms or create semantics and build a RDF/RDFS Knowledge Graph. Detecting inconsistent labels would be a simple task in that graph, from which we would then be able to add axioms and rules that would help detect more 'semantic' errors that the IE tools did not detect. Furthermore, we can evaluate each of the entities extracted for completeness or determine if the instances are uniformly extracted.
Priority	This would be developed and improved over the class timeline
Stakeholders	Development Team: The stakeholders for this use case are the four developers: Amar

	Viswanathan, Sabbir Rashid, and Ian Gross. The course instructors, Professor McGuiness and Professor Kendall, are mentors for the use case. Users: Ontology Engineers
Description	In this use case we convert the outputs of REACH (Mihai Surdeanu's Clulab) task, which is specified in the FRIES format. The extraction system performs IE on publications from PubMed and other biomedical domain related conferences. These outputs are present as events, entities, sentences, relationships, passages and contexts. 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. Since the output of these systems are rarely in any other form than XML documents, it is quite difficult for developers and users to figure out the *semantic* inadequacies of the IE system. As pertaining to a closed world assumption with a set of predefined rules for populating instances may be impractical, we can first check if certain labels of entities are incorrect using a predefined subset of possible rules. Given a set of labels and classes, we can use rules to figure out if these labels and classes are assigned inconsistently. **Illustrative Example :** An example rule could be that "CLINTON instanceOf PERSON" for a particular dataset. This statement is derived from the supporting ontology. "PERSON" is defined already, while "CLINTON" is inserted based on IE output. Now any "CLINTON" that is extracted as a "LOCATION" instance becomes a trigger for an anomaly. In the ontology we would specify that an individual cannot be both a PERSON and LOCATION. This kind of rule will be added as a disjointness axiom. The ontology can also be used to add other axioms to check boundary conditions, subtype relations etc. Our goal is to build such kind of axioms for the REACH outputs. In addition to checking for correctness, such a knowledge graph can also give quick summaries of entities, relationships and the events. This is achieved by writing simple SPARQL queries (select COUNT(*)...) and sending them to the designated triple store (Virtuoso)
Actors / Interfaces	The primary actors of the service are the Evaluation Interface and End User. Secondary actors may include: - Ontology Reviewers - Students - IE Developers - Task Evaluators - Ontology Engineers - Data Source (Knowledge Graph) - IE Toolkit
Preconditions	Before operation of the KGES tool, there must exist outputs from an IE toolkit. Furthermore, an ontology (created as part of the system) is required, which would be used to drive the evaluation. This includes terms extracted from the IE Output, as well as predetermined schema relations.
Postconditions	Inconsistencies found will be reported in a document and displayed by a

	visualization tool. A basic set of rules that may define consistency are disjointness between individuals and/or concepts, disjointness between relations, misclassification of events/processes, misclassification of relations and misclassification of individuals and/or concepts. The inconsistency will either be displayed as a graph illustration, a list, or another form of visualization. The inconsistency will display: the name of the entity, what tags were given to the term, and what misclassifications were found.
Triggers	The user would log into the system, choose a knowledge graph and then look at which rules are violated. These violations will be reported as a histogram, a table, or a visual graph with marked inconsistencies. The histogram will show the number of times a specific rule is broken. The table will show all the reported inconsistencies in a table that has the following categories: term(s) extracted, tag(s), relation (if inconsistency is between two terms), and the inconsistency. A visual graph will showcase the supporting ontology, where nodes are represented by concepts/individuals and edges are represented by the relation between nodes.
Performance Requirements	Java8 compliant systems, 8 GB of RAM for processing, 3.2 GB disk storage.
Assumptions	Output Knowledge Graph Generated from IE toolkit contains errors or inconsistencies.
Open Issues	Can inconsistencies be corrected automatically? How does one write rules for this?

III. Usage Scenarios

Scenario 1: Jack

Jack is a QA engineer working on a biomedical recommendation system, which seemingly is not working well. He thinks it might be due to some inconsistencies in knowledge base, which contains a graph of connecting concepts and relations on which the recommendations are based off. There are several rules he believes are true including “a drug is not a disease,” which he adds to an ontology containing his knowledge about medicine. The ontology contains rules pertaining to the disjointness of a set of biomedical concepts, as well as biomedical term definition and hierarchy structure. The ontology is used to support the Graph Evaluation system, into which he imports his knowledge graph. Once evaluation completes (the process of which is described in the following scenario), the system provides multiple processes to view the various inconsistencies discovered. He looks at the histogram, which displays the number of inconsistencies per type of rule. The overall accuracy response of the graph is given to be 60% (where accuracy is defined as the amount of terms involved in inconsistencies / the total number of terms). Inconsistencies refer to the rules defining disjointness of a particular class or instance, or disjointness between a class or instance. Terms in this context can be referred individually to the set of entities, relations, or processes. He then submits a SPARQL query asking about which specific rule contributes most to the percentage of inaccuracy, which reveals that the rule “a drug is not a disease” contributes to 35% of inaccuracy. Therefore, he discovers the main cause of the inconsistencies. This information could also be broken down in a histogram, showcasing which inconsistency rules were attached and if more than one inconsistency rules were attached to the same relation.

Scenario 2: Rachel

Rachel is a computer science student at Rensselaer Polytechnic Institute. She wants to use the Knowledge Graph Evaluation System to make sure a pubmed document is using language in a correct manner. The text document is submitted to the application, which is interpreted by the IE. The Knowledge Graph Generator uses the three created XMLs generated by the IE, which contains entity-mentions, relation-mentions, and event-mentions, to generate an RDF knowledge graph. The Evaluation Service is utilized to compare the inconsistency ontology and the previously generated RDF graph. The output will provide what particular concepts/individuals fell under an inconsistency rule, present the count of terms that were mapped, and value/type mismatch and domain/range mismatch. Suggestions of mappings to SIO or supporting ontology concepts may also be provided. This can be done using one of three metrics for determining the best match in a set of extracted relations pertaining to a single concept. The first method takes a probabilistic count based approach, where the number of documents that contain each relation are tallied, with the greatest value pertaining to the most likely classification. Another approach is based on a similarity metric, where the words in the description and label of a proposed term is compared with the original concept. Finally, an exhaustive approach may initially include all proposed possible concepts and relations in the generated KG, but then check rules in the ontology to determine if inconsistencies appear. From these results, Rachel is able to determine incorrect grammar that appears in the news article.

Scenario 3: Mary

Mary is a knowledge graph evaluator from 3M. She has been working on a knowledge graph of their various products and assets. She would like to get a summary of how many individuals in a product catalog have a relation to a specific adhesive concept type. This relation can be anything from a hasAdhesiveProperty to hasManufacturingLocation. This was brought about because there has been a complaint on the adhesive property, so they need to track which products use the adhesive. Mary is using the KGES because her own SPARQL service did not show all the expected products (like the adhesive being made in a specific factory that she already knows produces it and didn't show in SPARQL query) and she needs to correct these inconsistencies before reporting to the manufacturer. Mary would submit her RDF graph to the application. The application would utilize its evaluation service to compare inconsistencies via the ontology. To work with her RDF graph, Mary has updated the ontology with entity relations for adhesive products. Once complete, Mary can issue a SPARQL query provided on the visualization tool. In this case, she wants to check for the product: 3M™ Optically Clear Adhesive 9483. The service provides a listing of all products that utilize this adhesive. While browsing the SPARQL service results, the system informs Mary of an inconsistency in regards the adhesive property. Similar product classes were defined with different properties than the others, causing them to not show up in the search. This shows up as a mismatch inconsistency. The KGES provides a sample solution, which is to maintain a similar property hierarchy for similar classes.

Scenario 4: Mark

Mark is an ontology engineer at a biomedical consulting company. His team has created an ontology based on various pubmed documents related to their various projects and other related terms. The team has noticed a few odd connections in the knowledge graph. Mark has been tasked with figuring out the correctness based on the ontology. He submits his RDF knowledge graph of pubmed data to the application, which provides inconsistency results. Based on the visualization from the KGES, Mark is able to make some minor adjustments to the ontology to better align with the relation between terms as specified by the ontology. He looks at the generated histogram of accuracy for each provided term. Mark may notice anomalies of a particular class. He can then check the system to see which rules it is violating so that this can be corrected. The system also provides the general statistic of what percentage of the entire knowledge graph matches incorrect connections as described in the ontology.

Scenario 5: Alex

Alex is a mobile application developer, and he is working on an application that is a QA system for medical system.

He is attempting to determine which database he would like to use for his application. The possible data source candidates include: “Reach output on the 1K papers from the summer 2015 Big Mechanism DARPA evaluation” and “Reach output on the Open Access subset of PubMed”. Because the data sets are similar in content, he wants to determine which data set based on consistency, leading to a more reliable application. He submits each data source to the KGES. A supporting knowledge graph is generated for each data source. Alex looks through the visualization tool results. A list is provided for each data source showcasing the various inconsistencies discovered and the number of inconsistencies by category and importance. Based on these results, Alex decides that the “Reach output on the Open Access subset of PubMed” has less inconsistency and to use this as his data source. He also corrects the inconsistencies provided based on the results of the KGES.

IV. Basic Flow of Events

Narrative: *Often referred to as the primary scenario or course of events, the basic flow defines the process/data/work flow that would be followed if the use case were to follow its main plot from start to end. Error states or alternate states that might occur as a matter of course in fulfilling the use case should be included under Alternate Flow of Events, below. The basic flow should provide any reviewer a quick overview of how an implementation is intended to work. A summary paragraph should be included that provides such an overview (which can include lists, conversational analysis that captures stakeholder interview information, etc.), followed by more detail expressed via the table structure.*

In cases where the user scenarios are sufficiently different from one another, it may be helpful to describe the flow for each scenario independently, and then merge them together in a composite flow.

Basic / Normal Flow of Events			
Step	**Actor (Person)**	**Actor (System)**	**Description**
1	**User**		Launches the application
2		KnowledgeGraphApp, IE, Knowledge Graph Generator	The provided text documents are converted to three XML documents via the IE service. Information Extraction (IE) toolkits uses Entity, Event and Relationship Extraction procedures to extract information in the form of entities (individuals and concepts), events, and relationships, respectively. The three XMLs are converted to an RDF via the Knowledge Graph Generator by parsing the XML and associating entities and events using the extracted relations. Note that such a knowledge graph will likely contain inconsistencies.
4		KnowledgeGraphApp, EvalutionService	The Evaluation Service compares inconsistency ontology with RDF graph to create results to display to User.
5		KnowledgeGraphApp, DisplayStats, Check for inconsistencies	Statistics and inconsistencies are provided to user.
6	User		User may query for specific inconsistencies.

Subordinate Diagram #1 - Text Document to XML to RDF graph			
Step	**Actor (Person)**	**Actor (System)**	**Description**

1	**User**		The user gathers a text document.
2	User		Launch Application
4	**User**		The user submits the text document to the application
5		KnowledgeGraphApp, IE	The information extraction service splits the text document into three XML documents: Entity, Event and Relationship Extraction procedures.
6		KnowledgeGraphApp, Knowledge Graph Generator	The resulting documents are in the XML/JSON format. These documents are are converted to an RDF graph via the Knowledge Graph Generator. This will utilize a reference ontology.

Subordinate Diagram #2 - RDF Evaluation			
Step	**Actor (Person)**	**Actor (System)**	**Description**
1		KnowledgeGraphApp, EvalutionService	The Evaluation Service is utilized to compare the inconsistency ontology and the previously generated RDF graph.
2		KnowledgeGraphApp	Output is a visualization of inconsistencies.

Subordinate Diagram #3 - Inconsistency Display			
Step	**Actor (Person)**	**Actor (System)**	**Description**
1		KnowledgeGraphApp, DisplayStats, Check for inconsistencies	Visualization is provided to the user. Retrieves Statistics for the extracted graph such as number of entities, relations, events, number of populated instances, percentage of correctness compared with baseline systems.
2	**User**		The user drills down on specific statistics for instances of classes, applies existing rules to check for inconsistencies.
3		KnowledgeGraphApp	Displays histogram of inconsistencies with accuracy percentages

V. Alternate Flow of Events

Narrative: *The alternate flow defines the process/data/work flow that would be followed if the use case enters an error or alternate state from the basic flow defined, above. A summary paragraph should be included that provides an overview of each alternate flow, followed by more detail expressed via the table structure.*

Alternate Flow of Events #1 - Input Error			
Step	**Actor (Person)**	**Actor (System)**	**Description**
1	**User**		Launches the application
2		KnowledgeGraphApp	Extracted graph is unextractable (Parsing Failure). Reports error to user and application exits before

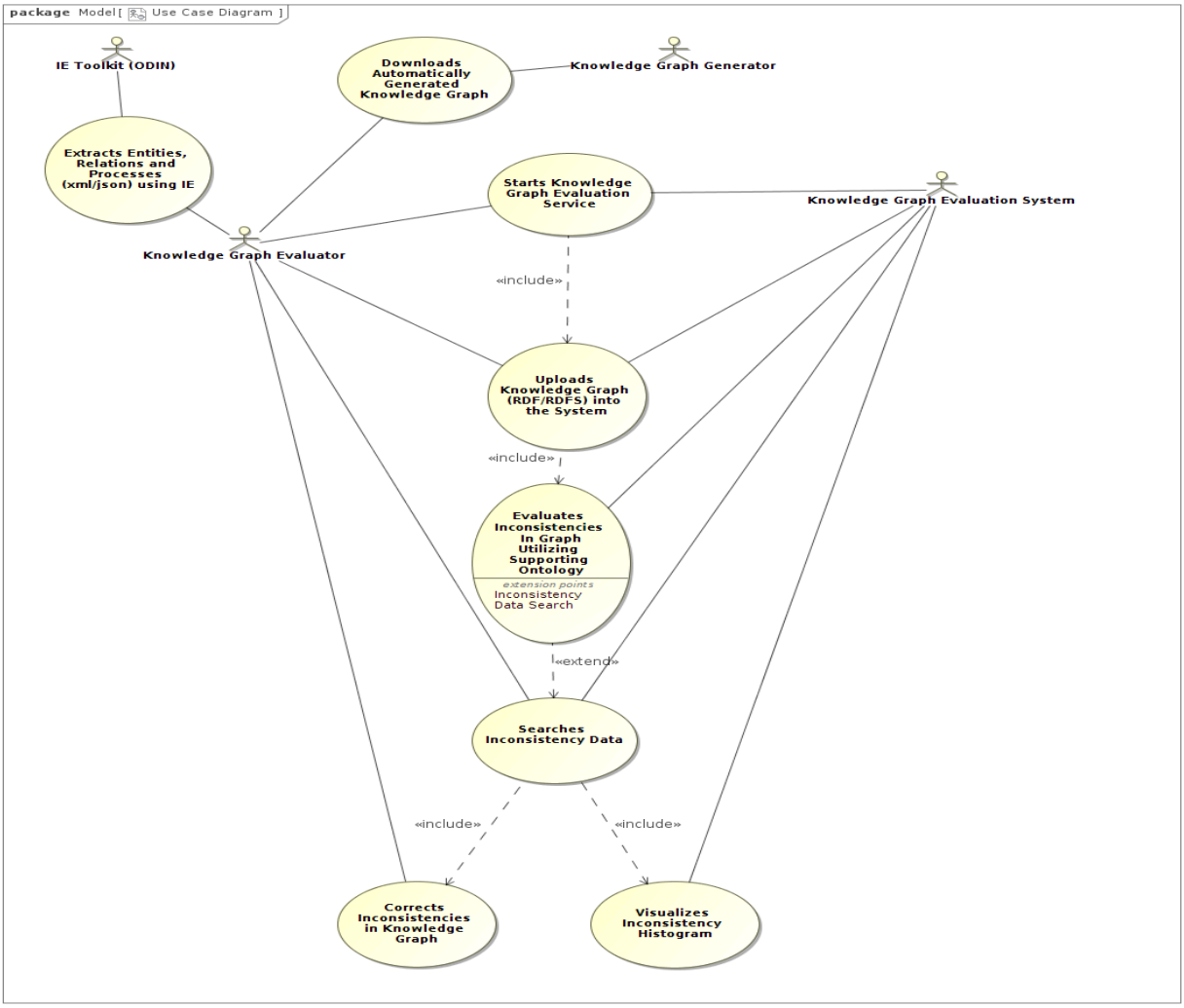
			DisplayStats and Check for inconsistencies runs.

Alternate Flow of Events #2 - External Knowledge Graph			
Step	**Actor (Person)**	**Actor (System)**	**Description**
1	**User**		Launches the application
2	**User**		User submits personal RDF Graph to check for Inconsistencies.
3	**User**		User updates supporting ontology to support the terminology in submitted RDF Graph.
4		KnowledgeGraphApp	Inconsistencies generated and application continues as normal behaivor.

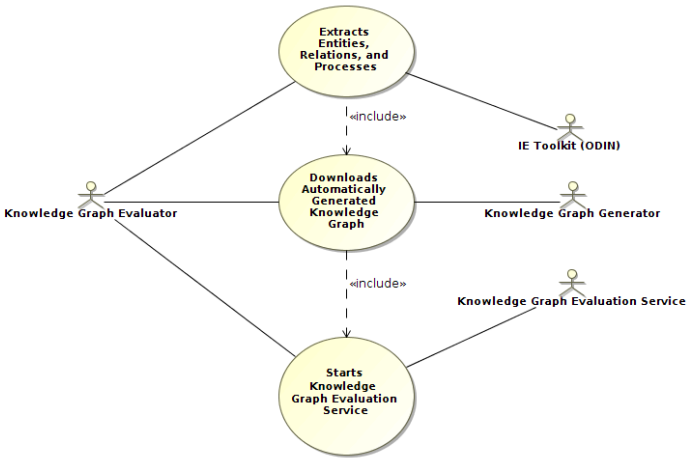
VI. Use Case and Activity Diagram(s)

Provide the primary use case diagram, including actors, and a high-level activity diagram to show the flow of primary events that include/surround the use case. Subordinate diagrams that map the flow for each usage scenario should be included as appropriate

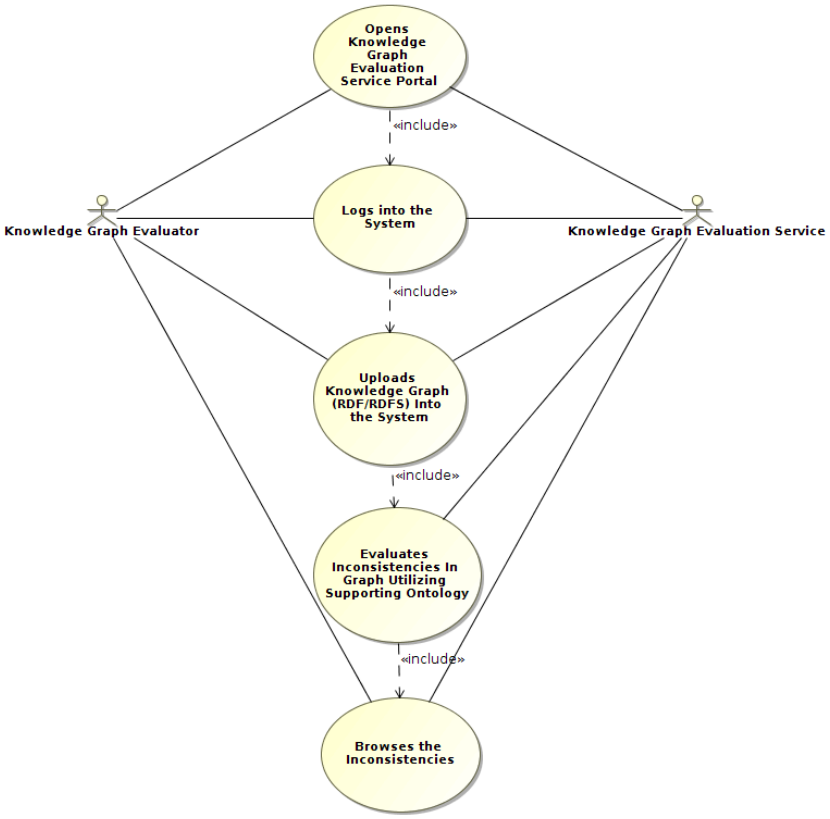
Primary Use Case:



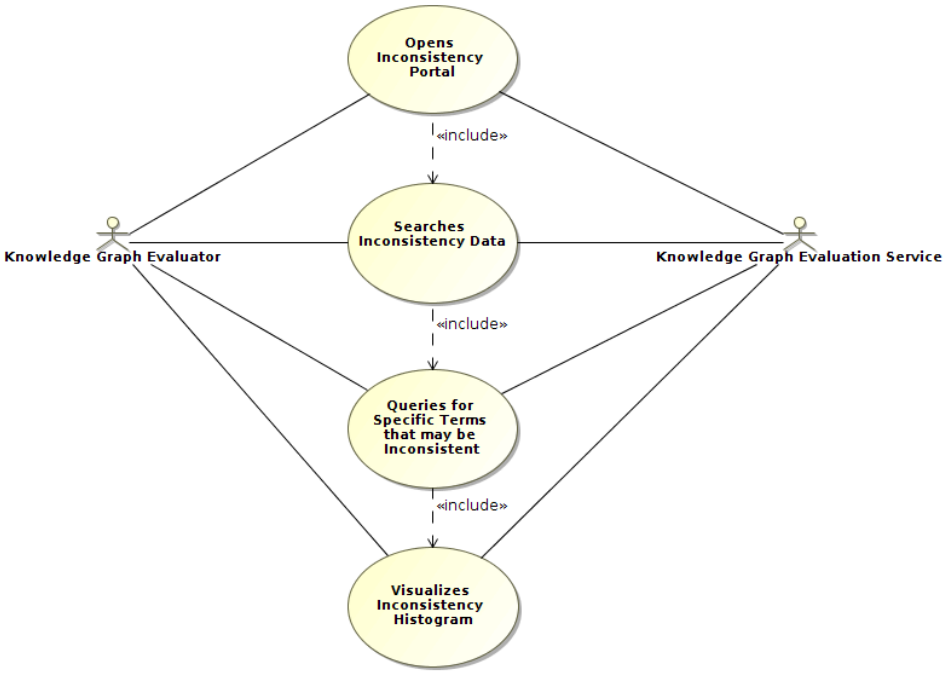
Knowledge Graph Generation:



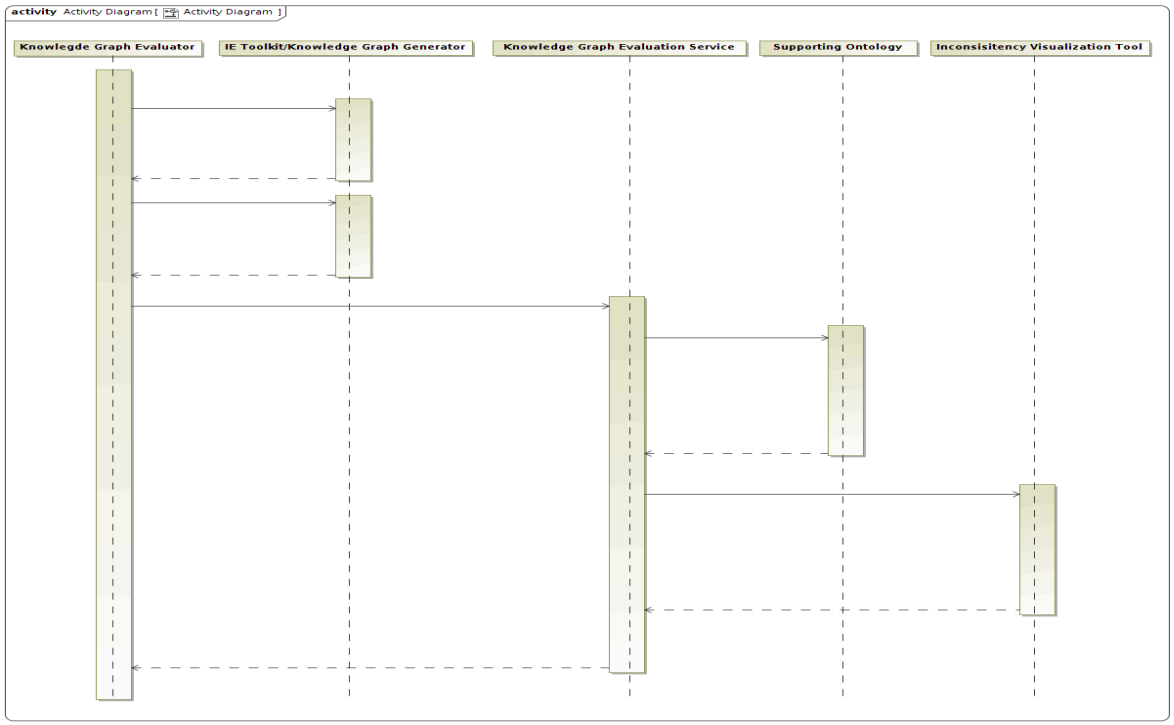
Upload Knowledge Graph to Evaluation Service:



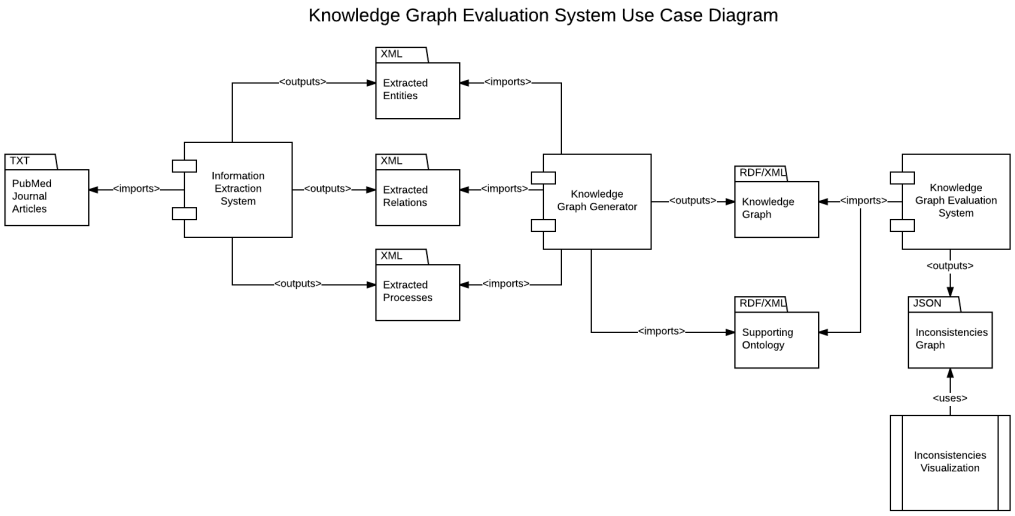
Inconsistency Visualization Use Case:



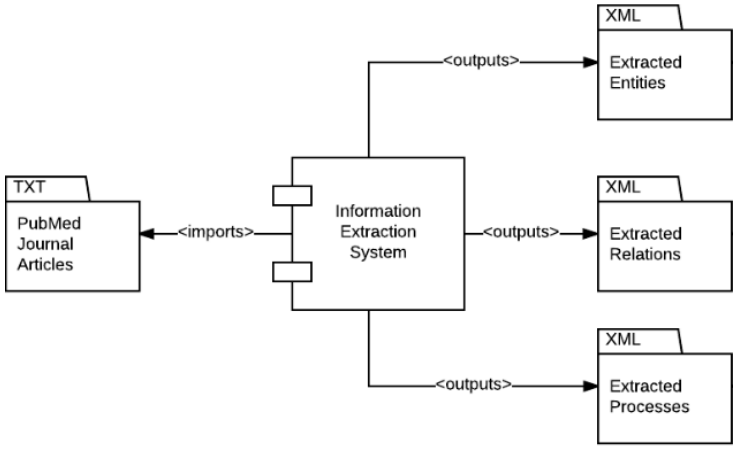
Activity Diagram:



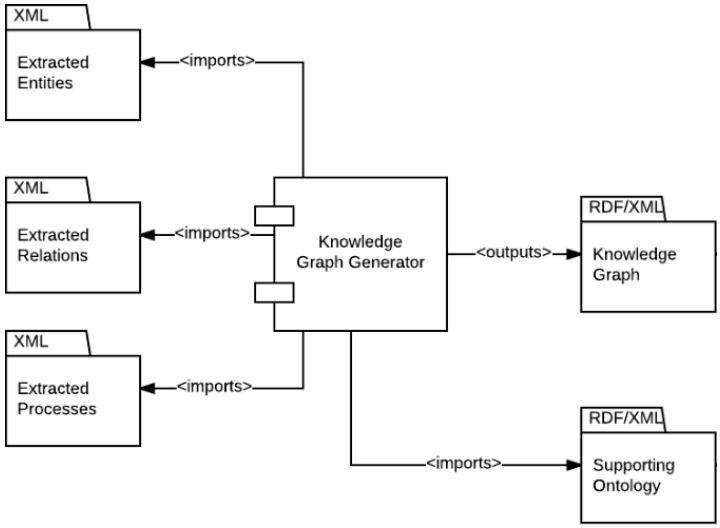
Design Diagrams:



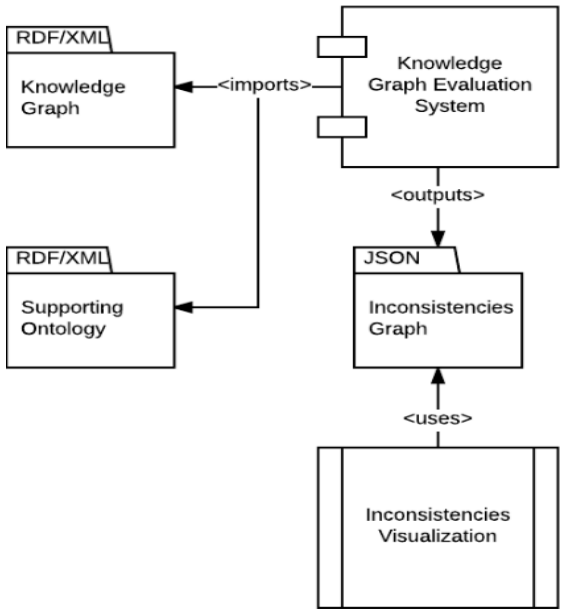
Information Extraction System:



Knowledge Graph Generator:



Knowledge Graph Evaluation System:



VII. Competency Questions

1) Given a Knowledge Graph from PubMed documents, is it a consistent graph based on the rules defined in the supporting ontology?
    - **Ans:** If there are no violations of any instances to the basic rules, then the system would report a

consistent graph. For example, if there are no multiple labels for the disjoint classes, then the graph is a consistent graph. A basic set of rules that may define consistency are disjointness between entities, disjointness between relations, misclassification of events/processes, misclassification of relations and misclassification of entities.

- Ontology Example: Given the PubMed documents, the IE classifies the information and it's compared to the supporting ontology. In this case, let's say that it finds a total of 40 inconsistencies. The method of how these inconsistencies are classified are not important for this case and will be covered in the other competency questions. The inconsistencies found will be displayed to the user, who can find out why they were considered inconsistent. If the IE found 500 terms, the document would come back as 8% inconsistent.
- Example Rules:
    i) 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 know to be a sedative type of drug. Librium also came back as isDrugType 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.
    ii) Disjointness between relations: A rule of this nature may concern when an individual has an incorrect property relation with another individual. For example, a disease with no cures called Schizophrenia is listed as a term in the supporting ontology. Another individual named Chlorpromazine (a medicine) has a property IsCureForDisease, which claims to be related to Schizophrenia. Because the disease has no expected cures, this is considered a disjoint between two terms. One relation says that it shouldn't have any cures, while the other says that it is a cure for the other. The relations are inconsistent with one another. Therefore, one relation must be wrong.
    This might also be a misclassification. Chlorpromazine may belong under a relation such as IsTreatmentForDisease, but the KGES may not know that information.
    iii) Disjointness between entities: Two individuals or concepts might be expected to have a relation to one another, yet the KGES might catch when no connection was made. One example could be an experiment on Alzheimer's Disease (an individual of the concept, clinical trial). A property could be the location of the trial (let's say Albany Medical Center). The location has several doctors working at the location. The trial has no doctor individual attached to the trial, but it is assumed that one of the individuals probably worked on this trial. Therefore, a connection should have been made from a doctor to this trial, and KGES will acknowledge that.

2) What conditions are required for a Misclassification of an Entity to be detected by the system?
    - **Ans**: If the Supporting Ontology contains the correct or expected classification of the term in question, then this misclassification may be detected.
    - Ontology Example: Cotinine is misclassified as an Environmental Phenol, although it is actually a Tobacco Metabolite. If the supporting ontology included a hierarchy tree of analyte relations, then the Knowledge Graph Evaluator could determine that a misclassification has occurred.
    - Consistency Rule: Misclassification of entity. TOBACCO METABOLITE is a type of molecule that acts an intermediary for tobacco. It has subclasses Cotinine and 3-butadiene mercapturic acid. In this case, these terms are classified incorrectly as a subclass of PHENOL, which is an aromatic organic compound. Cotinine and 3-butadiene mercapturic acid do not match the conditions to be marked as a PHENOL. Additionally, they actually fit under what is defined as a TOBACCO METABOLITE. Therefore, misclassification occurred because the entities didn't match its parent classification, while fitting the role for another.

Example Table Results: Query extraction of PHENOLS		
Term Extracted	**Tags**	**Inconsistencies**
Benzophenone-3	PHENOL	N/A
6-Tetrachlorophenol	PHENOL	N/A
2-Tetrachlorophenol	PHENOL	N/A
Cotinine	PHENOL	Actually a TOBACCO METABOLITE
3-butadiene mercapturic acid	PHENOL	Actually a TOBACCO METABOLITE

3) Let the rule in ontology be if there is a relation "catalysis accelerate" between two objects classified as "catalyst" and "chemical reaction", then there should not be relation "slow down" between two objects. In other words, the two relation are disjoint with each other. The question is "Are all the relations extracted consistent with this rule?"
    - **Ans:** If the relations extracted contains such inconsistent pair of relations, it will offer an visualization of the inconsistent disjoint relations just like the picture in next question. The OWL representation for these rules would be the intersection of "catalyst" and "chemical reaction", where the property "relation" is not equal to "slows down".
    - Ontology Example:

Example Table Results: Query extraction of CATALYST						
No.	**Term Extracte**	**Tags**	**Term Extracted**	**Tags**	**Relation**	**Inconsistencies**
1	MnO2	CATALYST	H2O2 decomposition	chemical reaction	catalysis accelerate	Inconsistent with relation No.2
2	MnO2	CATALYST	H2O2 decomposition	chemical reaction	slows down	Inconsistent with relation No.1

4) Which entity extractions are mapped to disjoint types?
    - **Ans:** KGES will use disjointness rules defined in a Supporting Ontology to determine if an inconsistency had occurred. The disjointness in question is the relationship between terms.

    One instance defines the relation:
<ment-PMC524479-UAZ-r1-524479-55-73> rdfs:label <P200> kgcs:hasMentionType <Protein> kgcs:fromSentence <sent-PMC524479-UAZ-r1-524479-55>

    Another instance defines the relation:
<ment-PMC524479-UAZ-r1-524479-55-74> rdfs:label <P200> kgcs:hasMentionType <Site> kgcs:fromSentence <sent-PMC524479-UAZ-r1-524479-55>

    The two individuals extracted from the same sentence have different values for kgcs:hasMentionType . If the Supporting Ontology includes the disjointness rule: disjointWith(Protein,Site) these individual entity mentions will trigger the corresponding disjointness inconsistency. This example assumes that a protein cannot be a site, which may not necessarily be true. However, it is possible to

consider more obvious cases, such as disjointWith(Site, Family) .

5) What is Event Disjointness? Give an example of this Inconsistency type for an Entity Mention.
   - **Ans:** Event Disjointness is defined as the inconsistency where the extracted events have an incorrect relationship to one another. In this case, we will be looking at an Entity Mention called GDNF protein. This term was extracted by the Information Extraction tool. This instance of GDNF protein was referenced as it goes through the methylation process. However, an additional instance of the GDNF protein was extracted and referenced that it goes through the dephosphorylation process. If the processes, methylation and dephosphorylation, are known to be disjoint in the supporting ontology, an inconsistency will occur.

6) What is an example of when the information is extracted incorrectly? (In terms of the phrase extraction)
   - **Ans**: When the Information Extraction tool extracts only part of a necessary entity mention boundary, causing a new entity mention to appear, this is known as Label Mapping Inconsistency. In other words, the system extracted a word without taking into account the context of the word in the overall sentence. Let's the following phrase as an example: "cholesterol going through hydroxylation". The overall phrase should have extracted cholesterol as a 'protein modification' event. However, the extractor instead returns a cholesterol as a protein entity mention due to the extraction boundary, which was missing the crucial term modification. This entity mention would classify an inconsistency because the term was extracted incorrectly. This can be determined by factors in the sentence structure. In this case, hydroxylation required that the process must have at least one chemical that it acts on. With no provided chemical, the system would be able to detect the incorrect Label Mapping Inconsistency.

7) 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?
   - **Ans**: This is an example of the general question: what entity mentions exist of a given label? To get these answers, a SPARQL query is issued to get results on where CTLA-4 was mentioned. A list of 86 Entity Mention results was displayed. Several sentences are showcased below, with the equivalent classes and the map type.

Example Table Results: CLTA-4 Entity Mentions

mention	mention label	From sentence	Mention type	Equivalent class	Equivalent class type
KGCS:ment-PMC555850-UAZ-r1-555850-176-515	CTLA-4	KGCS:sent-PMC555850-UAZ-r1-555850-176	protein	KGCS:protein-ment-PMC555850-UAZ-r1-555850-176-515	uniprot:P16410
KGCS:ment-PMC555850-UAZ-r1-555850-34-69	CTLA-4	KGCS:sent-PMC555850-UAZ-r1-555850-34	protein	KGCS:protein-ment-PMC555850-UAZ-r1-555850-34-69	uniprot:P16410
KGCS:ment-PMC546290-UAZ-r1-546290-140-421	CTLA-4	KGCS:sent-PMC546290-UAZ-r1-546290-140	protein	KGCS:protein-ment-PMC546290-UAZ-r1-546290-140-421	uniprot:P16410
KGCS:ment-PMC546290-UAZ-r1-546290-34-98	CTLA-4	KGCS:sent-PMC546290-UAZ-r1-546290-34	protein	KGCS:protein-ment-PMC546290-UAZ-r1-546290-34-98	uniprot:P16410

8) 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?

Example Table Results: ATP Event Mentions

Event mention	Event mention label	From sentence	Mention type	Mention subtype	Entity mention labels
KGCS:evem-PMC516774-UAZ-r1-516774-85-18	ATP	KGCS:sent-PMC516774-UAZ-r1-516774-85	activation	negative-activation	UO126 (25 muM), ATP
KGCS:evem-PMC546331-UAZ-r1-546331-156-54	ATP	KGCS:sent-PMC546331-UAZ-r1-546331-156	protein-modification	hydrolysis	cat, RecA, ATP, muM min -1, pH
KGCS:evem-PMC546331-UAZ-r1-546331-5-78	ATP	KGCS:sent-PMC546331-UAZ-r1-546331-5	protein-modification	hydrolysis	cat, ATP
KGCS:evem-PMC552968-UAZ-r1-552968-69-24	ATP	KGCS:sent-PMC552968-UAZ-r1-552968-69	complex-assembly	null	MSH2-MSH6, Cd, ATP

9) What events and entities are extracted from a given sentence? What are their mention types and subtypes?

Example Table Results: Event and Entity Mentions from a single Sentence

Sentence	Entity mention label	Entity mention type	Event mention labels	Event mention type	Event mention subtype
KGCS:sent-PMC512291-UAZ-r1-512291-95	epithelial cells, mouse, HIP, Ptc-1, membrane	celltype, species, protein, cellular-component	Ptc-1	activation	negative-activation
KGCS:sent-PMC544557-UAZ-r1-544557-51	alpha2 subunit	protein	alpha2 subunit	protein-modification	phosphorylation
KGCS:sent-PMC544557-UAZ-r1-544557-51	alpha2 subunit	protein	alpha2 subunit	protein-modification	dephosphorylation
KGCS:sent-PMC516018-UAZ-r1-516018-148	MBs, K8/18, MB	protein	MBs, K8/18	protein-modification	phosphorylation
KGCS:sent-PMC535350-UAZ-r1-535350-92	FAIM, GC B-cells, BCR, CD40L, GC, apoptosis, FASL	protein, celltype, bioprocess	FASL	activation	positive-activation
KGCS:sent-PMC516774-UAZ-r1-516774-100	ATP, phalloidin, diazoxide	simple-chemical	ATP	activation	positive-activation
KGCS:sent-PMC539278-UAZ-r1-539278-20	glutamate, NR1, CREB, MAP, rats, c-fos, striatal D-1 receptor, amphetamine,	site, protein, family, species, simple-chemic	NMDA receptor, NR1	protein-modification	phosphorylation

	NMDA receptor, NMDA, light	al			
KGCS:sent-PMC529456-UAZ-r1-529456-101	Fig, IRS1, serine 312, PMA, TSA	protein, site, organ	phosphorylation of IRS1 on serine 312	regulation	positive-regulation
KGCS:sent-PMC529456-UAZ-r1-529456-101	Fig, IRS1, serine 312, PMA, TSA	protein, site, organ	IRS1	protein-modification	phosphorylation

10) What are the number of distinct mention types extracted for an event mention?

Example Table Results: Event Mentions Distinct Types Count Event Mentions Distinct Types Count

Event mention	Mention types	Mention subtypes	Type count
JAK2	protein-modification, activation, complex-assembly, regulation, controller	phosphorylation, negative-activation, null, positive-activation, positive-regulation, autophosphorylation	5
EGF	activation, controller, regulation, complex-assembly, protein-modification	positive-activation, positive-regulation, null, negative-activation, phosphorylation, negative-regulation	5
Tat	protein-modification, activation, complex-assembly, theme	deacetylation, phosphorylation, positive-activation, null, acetylation, negative-activation	4
JNK	protein-modification, regulation, complex-assembly, activation	phosphorylation, positive-regulation, null, positive-activation, negative-activation	4

- How many different mention types for a label extracted from a single sentence?

Example Table Results: Event Mentions Distinct Types Count from a Sentence

Event mention	Mention types	Mention subtypes	sentence	Type Count
ATM	complex-assembly, controller, protein-modification	null, positive-regulation, autophosphorylation	KGCS:sent-PMC509302-UAZ-r1-509302-98	3
Tat	protein-modification, theme, complex-assembly	acetylation, null	KGCS:sent-PMC546329-UAZ-r1-546329-99	3
Gab1	complex-assembly, theme, activation	null, positive-activation	KGCS:sent-PMC534114-UAZ-r1-534114-7	3
Fu	protein-modification, theme, complex-assembly	phosphorylation, null	KGCS:sent-PMC545652-UAZ-r1-545652-27	3
GPCR	protein-modification, activation, regulation	phosphorylation, positive-activation, positive-regulation	KGCS:sent-PMC88976-UAZ-r1-88976-102	3
APOBEC3G	complex-assembly, activation	null, negative-activation	KGCS:sent-PMC520834-UAZ-r1-520834-158	2

alpha	complex-assembly, theme	null	KGCS:sent-PMC552954-UAZ-r1-552954-12	2

11) What are the number of distinct mention types extracted for an entity mention?

Example Table Results: Entity Mentions Distinct Types Count Event Mentions Distinct Types Count

Event mention	Mention types	Type Count
p57	family, protein, site	3
LEM-domain	family, site, protein	3
CTLA-4	protein, family, site	3
m262	site, cellline	2
S6	family, site	2
O3	site, simple-chemical	2

- How many different mention types for a label extracted from a single sentence?

Example Table Results: Entity Mentions Distinct Types Count from a Sentence

Entity mention	Mention types	Type count	Sentence
S10	site, cellline	2	KGCS:sent-PMC546156-UAZ-r1-546156-113
CENP-B	family, protein	2	KGCS:sent-PMC549403-UAZ-r1-549403-69
A549	cellline, site	2	KGCS:sent-PMC555846-UAZ-r1-555846-33
PLA2	site, family	2	KGCS:sent-PMC509240-UAZ-r1-509240-175
D1	site, cellline	2	KGCS:sent-PMC545049-UAZ-r1-545049-121
R8	site, organ	2	KGCS:sent-PMC548362-UAZ-r1-548362-42
v42	cellline, site	2	KGCS:sent-PMC540074-UAZ-r1-540074-2

VIII. Resources

In order to support the capabilities described in this Use Case, a set of resources must be available and/or configured. These resources include the set of actors listed above, with additional detail, and any other ancillary systems, sensors, or services that are relevant to the problem/use case.

Knowledge Bases, Repositories, or other Data Sources

Data	Type	Characteristics	Description	Owner	Source	Access Policies & Usage
REACH output in FRIES format on 1K	Json, nxml	e.g. – large extracted dataset of xml documents. The input sources is PubMed	Dataset is the output of clulab's information extraction system on PubMed documents This system extracts entities, events and relationships according to the FRIES format	Mihai Surdeanu	Mihai Surdeanu, Tom Hicks	Academic Usage

External Ontologies, Vocabularies, or other Model Services

Resource	Language	Description	Owner	Source	Describes/Uses	Access Policies & Usage
Uniprot	*RDFS*	Protein sequence and functional information		*http://www.uniprot.org/core/*	Used to link definitions of protein sequences that are extracted by the IE system.	Free and Open Source. Academic Use
SIO	RDFS	SIO is an integrated ontology that describes all kinds of objects,	Michel Dumontier	https://github.com/micheldumontier/semanticscience		Free and Open Source

		processes, attributes for the bio medical sciences				

Other Resources, Service, or Triggers *(e.g., event notification services, application services, etc.)*

Resource	Type	Description	Owner	Source	Access Policies & Usage
Virtuoso	Triple Store	Virtuoso will be the triple store of choice to store the RDF Graphs	For now we shall be hosting it on zen.cs.rpi.edu. But we can as well use any tetherless world server	***https://virtuoso.openlinksw.com/***	Free and open source version

IX. References and Bibliography

List all reference documents – policy documents, regulations, standards, de-facto standards, glossaries, dictionaries and thesauri, taxonomies, and any other reference materials considered relevant to the use case

FRIES-output-spec-v0.10_160502::
https://drive.google.com/drive/folders/0B_wQciWI5yE6RWFuSWlCWG8tQWM

1. Brank, J., Grobelnik, M., & Mladenic, D. (2005, October). A survey of ontology evaluation techniques. In *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)* (pp. 166-170).

2. Ding, L., Tao, J., & McGuinness, D. L. OWL Instance Data Evaluation.

3. Ding, L., Tao, J., & McGuinness, D. L. (2008, April). An initial investigation on evaluating semantic web instance data. In *Proceedings of the 17th international conference on World Wide Web* (pp. 1179-1180). ACM.

4. Dumontier, Michel, et al. "The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery." *Journal of biomedical semantics* 5.1 (2014): 14

5. Gómez-Péreza, A. OOPS!(OntOlogy Pitfall Scanner!): supporting ontology evaluation on-line.

6. Guo, M., Liu, Y., Li, J., Li, H., & Xu, B. (2014, May). A knowledge based approach for tackling mislabeled multi-class big social data. In *European Semantic Web Conference* (pp. 349-363). Springer International Publishing.

7. Hlomani, H., & Stacey, D. (2014). Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, 1-5.

8. McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000, April). An environment for merging and testing large ontologies. In *KR* (pp. 483-493).

9. McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). The chimaera ontology environment. *AAAI/IAAI*, *2000*, 1123-1124.

10. Pujara, J., Miao, H., Getoor, L., & Cohen, W. W. (2013). Knowledge graph identification.

11. Tao, J., Ding, L., & McGuinness, D. L. (2009, January). Instance data evaluation for semantic web-based knowledge management systems. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on* (pp. 1-10). IEEE.

12. Tao, J. (2012). *Integrity constraints for the semantic web: an owl 2 dl extension* (Doctoral dissertation, Rensselaer Polytechnic Institute).

13. Valenzuela-Escárcega, Marco A., Gus Hahn-Powell, and Mihai Surdeanu. "Description of the odin event extraction framework and rule language." *arXiv preprint arXiv:1509.07513* (2015).

X. Notes

There is always some piece of information that is required that has no other place to go. This is the place for that information.

- For any confusion on any of the terms listed in this document, whether it be related to the Information Extraction system, the extracted tags or ontology terms, please refer to the separate terms list for further information