



UNIVERSITÄT
PADERBORN

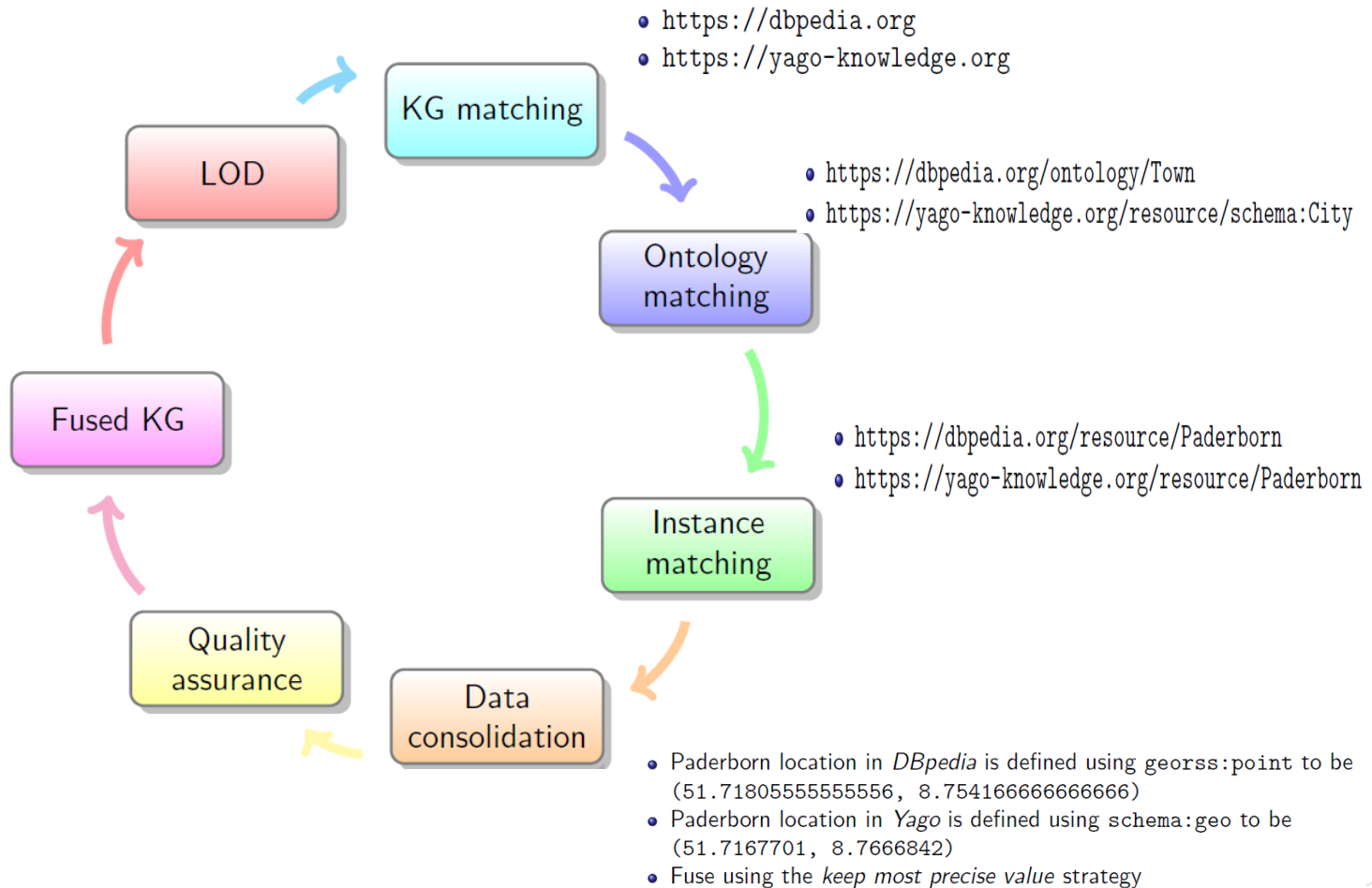


DICE GROUP

KNOWLEDGE GRAPH FUSION

ADVISOR : DR. RER. NAT. MOHAMED SHERIF

What is KG Fusion?



Overview

- ☐ KG Matching
- ☐ Ontology Matching
- ☐ Instance Matching
- ☐ Data Consolidation
- ☐ Summary

Overview

➔ ☐ KG Matching

➤ CHAITALI SUHAS BAGWE, RAVITEJA KANAGARLA

☐ Ontology Matching

☐ Instance Matching

☐ Data Consolidation

☐ Summary

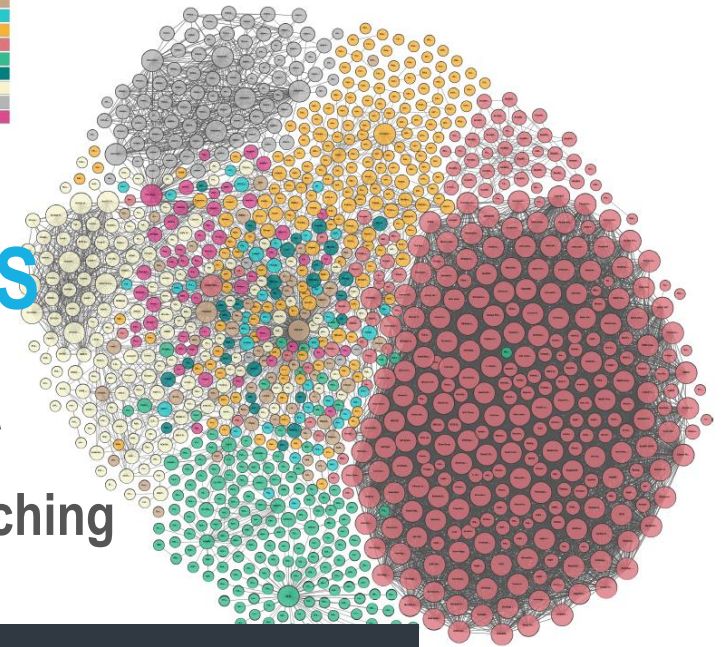
Goals

❑ Find out the similar Knowledge Bases

- ❑ Using LIMES for Linked Open Data Metadata for link matching
- ❑ Using various document matching techniques for datasets in Hobbit

❑ Create a KG Matching operator using DEER

❑ Give output to the next group i.e., Ontology Matching



Steps to match KGs using LIMES

- ❑ Collect the metadata from Linked Open Data
- ❑ Identify the metadata to be used for KG matching

```
{
  "wiktionary-dbpedia-org": {
    "_id": "wiktionary-dbpedia-org",
    "identifier": "wiktionary-dbpedia-org",
    "doi": "",
    "image": "",
    "links": [
      {
        "target": "dbpedia",
        "value": "25155"
      }
    ],
    "keywords": [
      "crossdomain",
      "linguistics",
      "localization-diagram-2014-08-30"
    ],
  },
}
```

```
"babelnet": {
  "_id": "babelnet",
  "identifier": "babelnet",
  "doi": "",
  "image": "",
  "keywords": [
    "access-web",
    "format-rdf",
    "ldl-2014",
    "lemon",
    "lexical-resources",
    "lexico",
    "linguistics",
    "llod",
    "void-sparql-endpoint"
  ],
  "links": [
    {
      "target": "dbpedia",
      "value": "27585374"
    }
  ],
}
```

Steps to match KGs using LIMES

❑ Create a DEER Operator

❑ Pass the collected metadata to LIMES framework via DEER Operator

```
public Configuration createLimeConfigurationFile(List<Model> models) throws IllegalArgumentException {  
    // Creating Limes configuration Object  
    Configuration conf = new Configuration();  
    // adding prefix  
    conf.addPrefix("ns1", "https://example.com/test#");  
    conf.addPrefix("owl", "http://www.w3.org/2002/07/owl#");  
    conf.addPrefix("rdfs", "http://www.w3.org/2000/01/rdf-schema#");  
  
    KBIInfo src = new KBIInfo();  
    src.setId("sourceId");  
    src.setEndpoint("jsontordfoutput.ttl");  
    //src.setEndpoint(String.valueOf(models.get(0)));  
    src.setVar("?o");  
    src.setPageSize(1000);  
    src.setType("TURTLE");  
    src.setRestrictions(new ArrayList<String>(Arrays.asList(new String[]{ "rs:ns1:dataset ?o" })));  
    src.setProperties(Arrays.asList(new String[]{ "ns1:keywords", "ns1:domain" }));  
    //src.setProperties(Arrays.asList(new String[]{ " " }));  
}
```

❑ Store the matched KGs in a RDF model

Limes Output

```
<http://wiktionary.dbpedia.org/sparql>
  <http://example.com/test#matches>
    <http://publications.europa.eu/webapi/rdf/sparql> , <http://wordnet.okfn.gr:8890/sparql/> , <
    http://wordnet.rkbexplorer.com/sparql/> , <http://babelnet.org/sparql/> , <http://demo.spraakdata.gu.se/
    ltlod/test/> , <http://linked-data.org/sparql> , <http://mione.nlp2rdf.org/sparql> , <http://zhishi.me/
    sparql> , <http://ld.panlex.org/sparql> .

<http://www.linklion.org:8890/sparql>
  <http://example.com/test#matches>
    <http://minsky.gsi.dit.upm.es/semanticwiki/index.php/Special:SPARQLEndpoint> , <
    http://ecs.rkbexplorer.com/sparql> , <http://purl.org/twc/hub/sparql> , <
    http://vocabulary.semantic-web.at/PoolParty/sparql/AustrianSkiTeam> , <http://meducator.open.ac.uk/
```




Steps to match KGs using Document Similarity

☐ Pre-processing hobbit datasets

- ☐ Find and split all literals in the dataset into 1-gram token
- ☐ Remove all stopwords, spaces, numbers and special characters
- ☐ Count the frequency of each token
- ☐ Sort the tokens according to their frequency and store them in a list

☐ Apply Document Similarities

- ☐ Matching each datasets against all datasets present
- ☐ Creating RDF model to store the matched similarities

Document Similarities Algorithms

- ☐ Jaccard Similarity
- ☐ Weighted Jaccard Similarity
- ☐ Dice Similarity
- ☐ TF-IDF with Cosine Similarity
- ☐ Bert Similarity

Output – Bert Similarity

```
<agrovoc_uniroma2_it.nt>
  <BertSimilarityOutput=0.3667376935482025>
    <data_nobelprize_org.nt> ;
  <BertSimilarityOutput=0.3763076663017273>
    <dbtune_org_bbc_peel_sparql.nt> ;
  <BertSimilarityOutput=0.5186800360679626>
    <dbtune_org_magnatune_sparql.nt> ;
  <BertSimilarityOutput=0.5870665907859802>
    <www_imagesnippets_com_sparql.nt> ;
  <BertSimilarityOutput=0.596704363822937>
    <data_ox_ac_uk_sparql.nt> ;
  <BertSimilarityOutput=0.5990902185440063>
    <dbtune_org_jamendo_sparql.nt> ;
  <BertSimilarityOutput=0.59983229637146>
    <www_orpha_net.nt> ;
  <BertSimilarityOutput=0.6346958875656128>
    <ldf_fi_ww1lod.nt> ;
  <BertSimilarityOutput=0.6368116140365601>
    <cdrewu_eagle-i_net_sparqler.nt> ;
  <BertSimilarityOutput=0.6752213835716248>
    <semanticweb_cs_vu_ni_verrijktkoninkrijk_sparql.nt> ;
  <BertSimilarityOutput=0.6857293844223022>
    <dbmi-icode-01_dbmi_pitt_edu.nt> ;
  <BertSimilarityOutput=0.6954165161651651>
    <onto_fel_cvut_cz_rdf4j-server_repositories.nt> ;
  <BertSimilarityOutput=0.6978609561920166>
```

Benchmarking

Dataset Pairs	A1	A2	A3	A4	A5	Mutual Agreement
(dbtune_org_jamendo, data_nobel)	✓	✓	✓	✗	✓	✓
(dbtune_org_jamendo, data_ox_ac)	✓	✓	✓	✓	✓	✓
(cdrewu_eagle, imagesnippets)	✓	✗	✓	✓	✓	✓
(cdrewu_eagle, dbtune_org_magna)	✓	✓	✗	✗	✓	✓

A1, A5 – Software Engineers

A2 – Mechanical Engineer

A3 – Civil Engineer

A4 – Electrical Engineer

Benchmarking

Algorithms	Precision Score	Recall Score	F1 Score
Tf-idf with Cosine Similarity	0.94	0.85	0.892
Jaccard Similarity	0.94	0.80	0.863
Weighted Jaccard Similarity	0.97	0.87	0.884
Dice Similarity	0.97	0.69	0.804
Bert Similarity	0.97	0.80	0.874
LIMES Framework	1	0.73	0.843

Future works

- ❑ More Document Similarities can be added
- ❑ Trying the approaches on bigger datasets (> 1GB)
- ❑ Using dedicated Knowledge Graph matching approach like
Tapioca

Overview

☐ KG Matching

➔ ☐ **Ontology Matching**

➤ SOWMYA KAMARTH RAMESH, KRISHNA MADHAV

☐ Instance Matching

☐ Data Consolidation

☐ Summary

Goals

- ☐ Integrating with DEER framework
- ☐ Implementation of another matching system (FCA)
- ☐ Benchmarking our Operator

Technologies

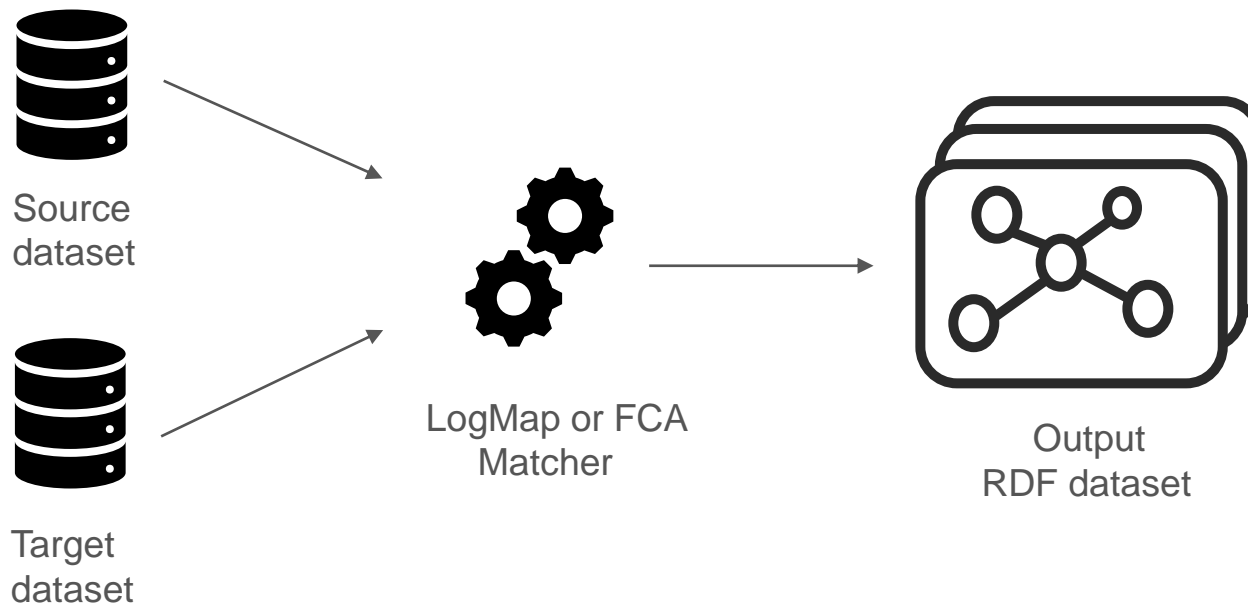
- ☐ JAVA
- ☐ DEER

Configuration File

```
1 :ontologyMatcher
2   a deer:OntologyMatchingOperator ;
3   deer:typeOfMap "Classes ";
4   deer:matching_Library "FCA";
5   fcage:hasInput :kgmatcher ;
6   .
```

```
1 :ontologyMatcher
2   a deer:OntologyMatchingOperator ;
3   deer:typeOfMap "Classes and ObjectProperty";
4   deer:matching_Library "LOGMAP";
5   fcage:hasInput :kgmatcher ;
6   .
```

Overview of our Operator



Ontology Matching Operator

- ☐ Dynamically fetching RDF datasets
- ☐ Accessing SPARQL endpoints
- ☐ Storing Ontologies in local file
- ☐ Generating Matched Mappings
- ☐ Reificated Output

Output

```
@prefix deer: <https://w3id.org/deer/> .

deer:Match_1  deer:found  [ a      <http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement> ;
                           <http://www.w3.org/1999/02/22-rdf-syntax-ns#object>
                           <http://eagle-i.org/ont/repo/1.0/Person> ;
                           <http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate>
                           deer:matchesWith ;
                           <http://www.w3.org/1999/02/22-rdf-syntax-ns#subject>
                           <http://eagle-i.org/ont/app/1.0/Person> ;
                           deer:ObjectEndPoint <http://tsu.eagle-i.net/sparqler/sparql> ;
                           deer:SubjectEndPoint <http://utep.eagle-i.net/sparqler/sparql> ;
                           deer:confidenceValue "0.6"
                           ].
```

Benchmarking

Dataset Pairs	A1	A2	A3	Mutual Agreement
NCI_overlapping_fma.owl, FMA_overlapping_nci.owl	✓	✓	✓	✓
SNOMED_overlapping_fma.owl, FMA_overlapping_snomed.owl	✓	✓	✓	✓
Conference.owl, Ekaw.owl	✓	✓	✗	✓

A1, A2 – Software Engineers

A3 – Mechanical Engineer

Benchmarking

Dataset Pairs	Matcher	Precision	Recall	F1
NCI_overlapping_fma.owl, FMA_overlapping_nci.owl	LogMap Matcher	0.94	0.92	0.93
	FCA Matcher	0.91	0.93	0.92
SNOMED_overlapping_fma.owl, FMA_overlapping_snomed.owl	LogMap Matcher	0.92	0.91	0.91
	FCA Matcher	0.93	0.86	0.89
Conference.owl, Ekaw.owl	LogMap Matcher	0.94	0.87	0.90
	FCA Matcher	0.92	0.87	0.89

Future Work

- ☐ Improving timeout for SPARQL query execution
- ☐ Working with dataset having triples pointing to invalid OWL files
- ☐ Handling bigger dataset

Overview

- ☐ KG Matching

- ☐ Ontology Matching

-  ☐ Instance Matching

 - KHALID KHAN, KHALID BIN HUDA

- ☐ Data Consolidation

- ☐ Summary

GOAL

- ☐ Implement a technique for Instance Matching
- ☐ Creating a DEER Operator

Technologies

- ☐ JAVA
- ☐ DEER
- ☐ LINES

Instance Matching Operator

❑ Named: InstanceMatchingOperator

❑ Highlights:

- ❑ Creating prefixes dynamically
- ❑ Calculate coverage of the properties
- ❑ Dynamically creating LIMES configuration
- ❑ Use Wombat simple algorithm
- ❑ Introducing Type-driven wombat simple

Configuration File

```
15
16 :matcher #add source and target and restriction(read from output of ontology matching group)
17 a deer:InstanceMatchingOperator ;
18 deer:coverage "0.70" ; #Coverage of a Property #propertyCount/TotalInstanceCount #Default=90%
19 deer:maxLimit "10"; #Maximum number of Properties #Default=3
20 deer:type "file"; #It can be "file" or "endpoint"
21 deer:source "data_nobelprize_org.nt";
22 deer:target "eu_dbpedia_org.nt";
23
24 deer:sourceRestriction [ deer:restrictionURI owl:class ;];
25 deer:targetRestriction [ deer:restrictionURI foaf:Person ;];
26
27 deer:tabuSourceProperty [ deer:propertyURI foaf:id ;];
28
29 fcage:hasInput :reader ;
30 .
31
```

Example

Data File source: dbmi-icode-01_dbmi_pitt_edu.nt

Source Restrictions: "?s rdf:type owl:Class"

Properties matched:

puob156:IAO_0000115
gefor191:hasOBONamespace
gefor335:id
w3200488:label

Data File target: agrovoc_uniroma2_it.nt

Target Restrictions: "?t rdf:type skos:Concept"

Properties matched:

w3200541:prefLabel
pudc237:created
pudc20:modified
w3200302:altLabel

wombat simple

OUTPUT:

<http://purl.obolibrary.org/obo/CHEBI_75958> <http://aims.fao.org/aos/agrovoc/c_28563> 0.6153846153846154

Benchmarking

- ❑ For benchmarking we slightly modified simple wombat
- ❑ Added a step to classify properties based on data type
- ❑ Use this classification while matching
- ❑ For example:

Vector Measure = [euclidean, manhattan]

String Measures = [jaccard, qgrams]

Temporal Measure = [tmp_predecessor, tmp_successor]

Product Name	Price	Manufacturing Date
Floppy disk	223	11-02-1997
Mouse	250	10-05-1998

String

Vector

Temporal

Label	Retail Price	Production Date
Disk	250	12-05-2000
Screen	500	19-01-2020

Benchmarking Result

Dataset	Execution Time Wombat Simple (ms)	Execution Time Type-Driven Wombat Simple (ms)	F-Measure Wombat Simple	F-Measure Type-Driven Wombat Simple	Accuracy Wombat Simple	Accuracy Type-Driven Wombat Simple
Amazon and Google	27821	16006	0.415	0.423	0.999	0.999
Abt-Buy	3969	2693	0.104	0.028	0.987	0.986
DBLP-ACM	64038	53567	0.889	0.900	0.999	0.999
Person 1	6985	6181	0.803	0.805	0.997	0.997
Restaurants	8944	8053	0.447	0.467	0.999	0.999

Future Work

- ❑ Implementing a more sophisticated Data type classifier. Use this classification while matching
- ❑ For a better performance use parallel programming

Overview

- ☐ KG Matching

- ☐ Ontology Matching

- ☐ Instance Matching

-  ☐ Data Consolidation

 - PHILIP COUTINHO DE SOUSA

- ☐ Summary

Goal

- ☐ Merge Data
- ☐ Flexible Usage

Technologies

- ☐ JAVA
- ☐ DEER

Consolidation Operator

❑ Name: ConsolidationOperator

❑ Idea:

- ❑ Source sameAs Target
- ❑ MatchableProperties
- ❑ Use FusionStrategies on Properties

❑ Highlights

- ❑ Easily expandable

Consolidation Strategies

Strategy	String	Integer	Date	Boolean	Fallback
Standard	Max	Avg	/	And	Take Source
Precise	Min	/	/	And	Take Source
Expertise Source	/	/	/	/	Take Source
Expertise Target	/	/	/	/	Take Target
Min/Max/Average /Union	Min/Max/Avg /Union	Min/Max/Avg / -	Min/Max/Avg / -	And	Take Source
Voting	Max Voting	Max Voting	Max Voting	Max Voting	Take Source

Configuration File

```
:consolidation
a deer:ConsolidationOperator ;
deer:sameAs "http://www.w3.org/2002/07/owl#sameAs"; # predicate that symbolizes equality
deer:entityName "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"; #predicate describes the given entitys
deer:sourceName "https://w3id.org/deer/datasetSource"; # predicate of the data Source
deer:targetName "https://w3id.org/deer/datasetTarget"; # predicate of the data Target
deer:addTarget "true"^^xsd:boolean ; # integrate unmatched from the target dataset
deer:namespaceForIntegration "https://w3id.org/deer/nameprefix/"; # prefix for integration of unmatched from target to
deer:provenanceProperty "https://w3id.org/deer/provenance";
deer:globalFusionStrategy "standard"; # standard, precise, expertise, [...]
deer:outputVariant "ttl"; # output file
deer:propertyFusion [ # Strategy by Propertyys, if not defined use globalFusion
  deer:propertyValue "http://xmlns.com/foaf/0.1/name"; ##property of the source Dataset
  deer:fusionStrategy "expertiseTarget"; ## same as globalstrategy
```

Example

Source:

```
<http://data.nobelprize.org/resource/laureate/448> <http://dbpedia.org/property/dateOfBirth> "1943-09-06"^^<http://www.w3.org/2001/XMLSchema#date> .  
<http://data.nobelprize.org/resource/laureate/448> <http://dbpedia.org/ontology/birthPlace> <http://data.nobelprize.org/resource/city/Derby> .  
<http://data.nobelprize.org/resource/laureate/448> <http://xmlns.com/foaf/0.1/givenName> "Richard J."^^<http://www.w3.org/2001/XMLSchema#string> .  
<http://data.nobelprize.org/resource/laureate/448> <http://www.w3.org/2002/07/owl#sameAs> <http://yago-knowledge.org/resource/Richard_J._Roberts> .  
<http://data.nobelprize.org/resource/laureate/448> <http://xmlns.com/foaf/0.1/name> "Richard J. Roberts"^^<http://www.w3.org/2001/XMLSchema#string> .
```

Target:

```
<http://sparql.cwrc.ca/ontologies/cwrc#26ad3610-a0bb-4e62-8fbc-d6be9ccbdf6-partof-327d5213ef>  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .  
<http://sparql.cwrc.ca/ontologies/cwrc#26ad3610-a0bb-4e62-8fbc-d6be9ccbdf6-partof-327d5213ef>  
<http://xmlns.com/foaf/0.1/name> " Richard J Lane"^^<http://www.w3.org/2001/XMLSchema#string> .  
<http://sparql.cwrc.ca/ontologies/cwrc#26ad3610-a0bb-4e62-8fbc-d6be9ccbdf6-partof-327d5213ef>  
<http://purl.org/dc/terms/lastName> "Richard J Lane"^^<http://www.w3.org/2001/XMLSchema#string> .
```

MatchableProperties:

```
<http://xmlns.com/foaf/0.1/name> - <http://www.w3.org/2001/XMLSchema#string> .
```

Example

<http://data.nobelprize.org/resource/laureate/448> <http://xmlns.com/foaf/0.1/name> "Richard J. Roberts"^^<http://www.w3.org/2001/XMLSchema#string> .


<http://sparql.cwrc.ca/ontologies/cwrc#26ad3610-a0bb-4e62-8fbc-d6be9ccbdf6-partof-327d5213ef>
<http://xmlns.com/foaf/0.1/name> " Richard J Lane"^^<http://www.w3.org/2001/XMLSchema#string> .

Strategy	Result
Standard	Roberts
Precise	Lane
ExpertiseSource	Roberts
ExpertiseTarget	Lane
Min/Max/Average/Union	Lane/Roberts/ - / RJ Roberts RJ Lane
Voting	Not Applicable -> Source -> Roberts

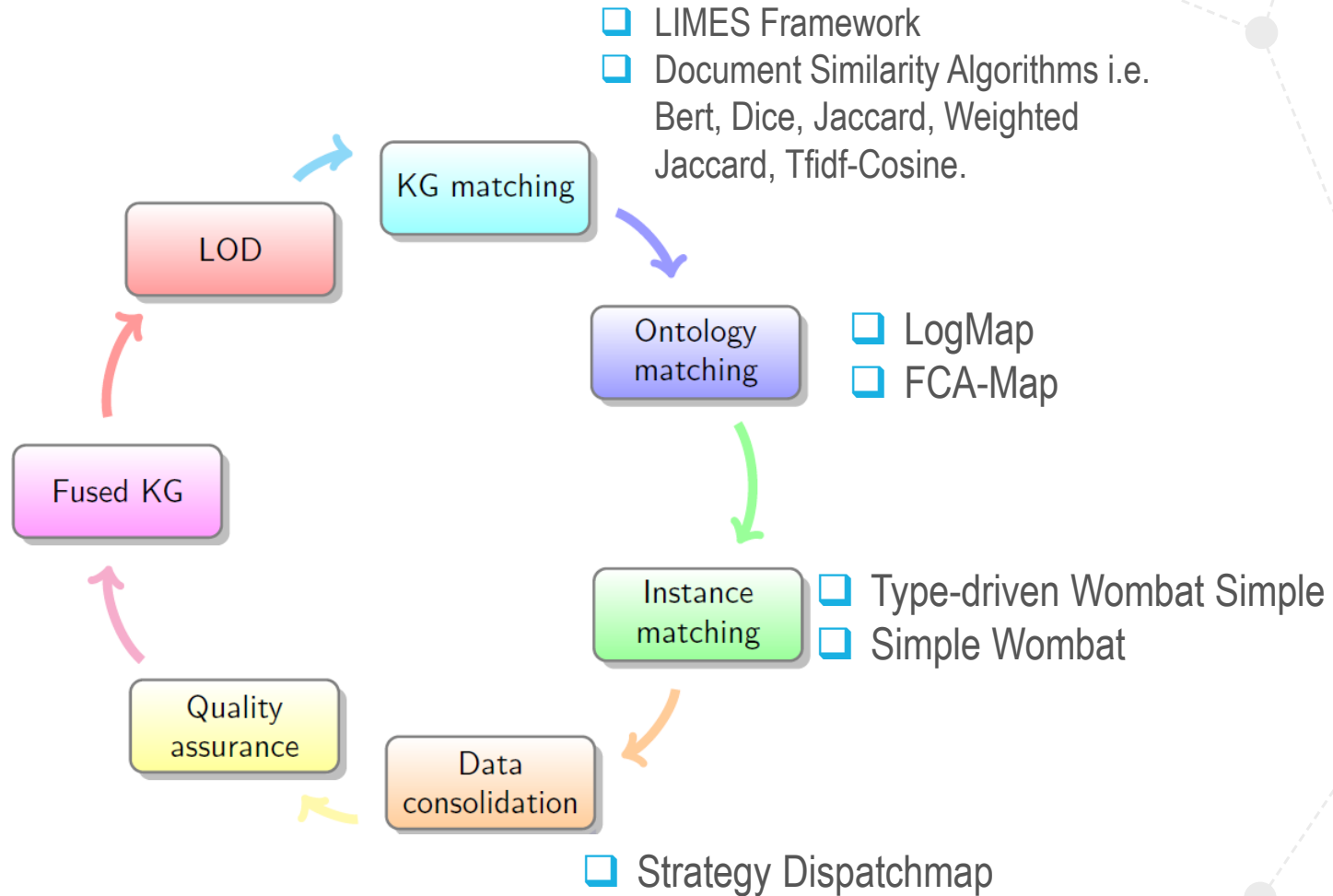
Future Work

- ❑ Matchable Properties from Previous Groups
- ❑ More Advance Fusion Strategies
- ❑ Different Output Formats
- ❑ New Output format
- ❑ Reification - Configuration

Overview

- ☐ KG Matching
- ☐ Ontology Matching
- ☐ Instance Matching
- ☐ Data Consolidation
-  ☐ Summary

Summary



THANK YOU FOR LISTENING

QUESTIONS?