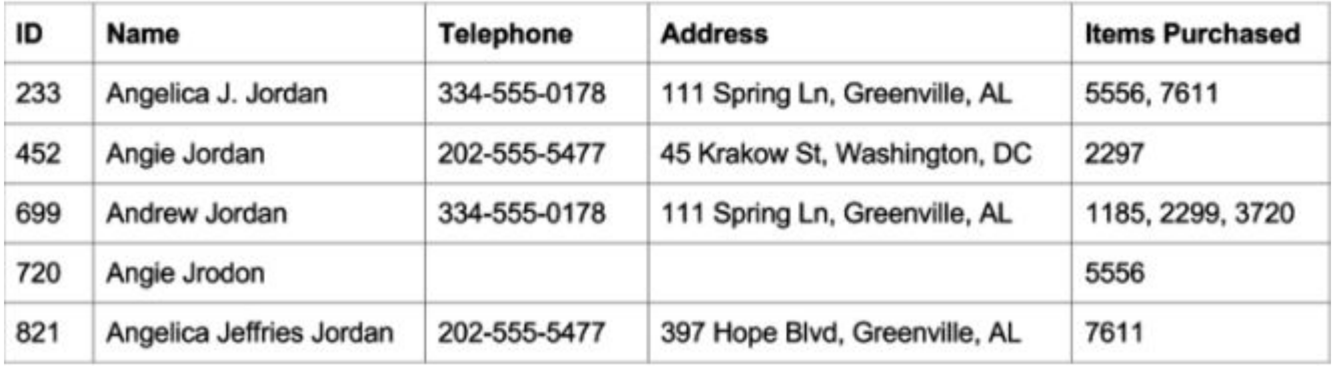
SEMANTIC BASED APPROACH FOR ENTITY MATCHING ON NOISY SEMI STRUCTURED DATA

Nikhil Acharya
Supervisor- Diego Collarana

ENTITY MATCHING

- Match data instances referring to same real world entities.
- Match records from 2 or more data sources
- The task is critical in data integration and data cleaning

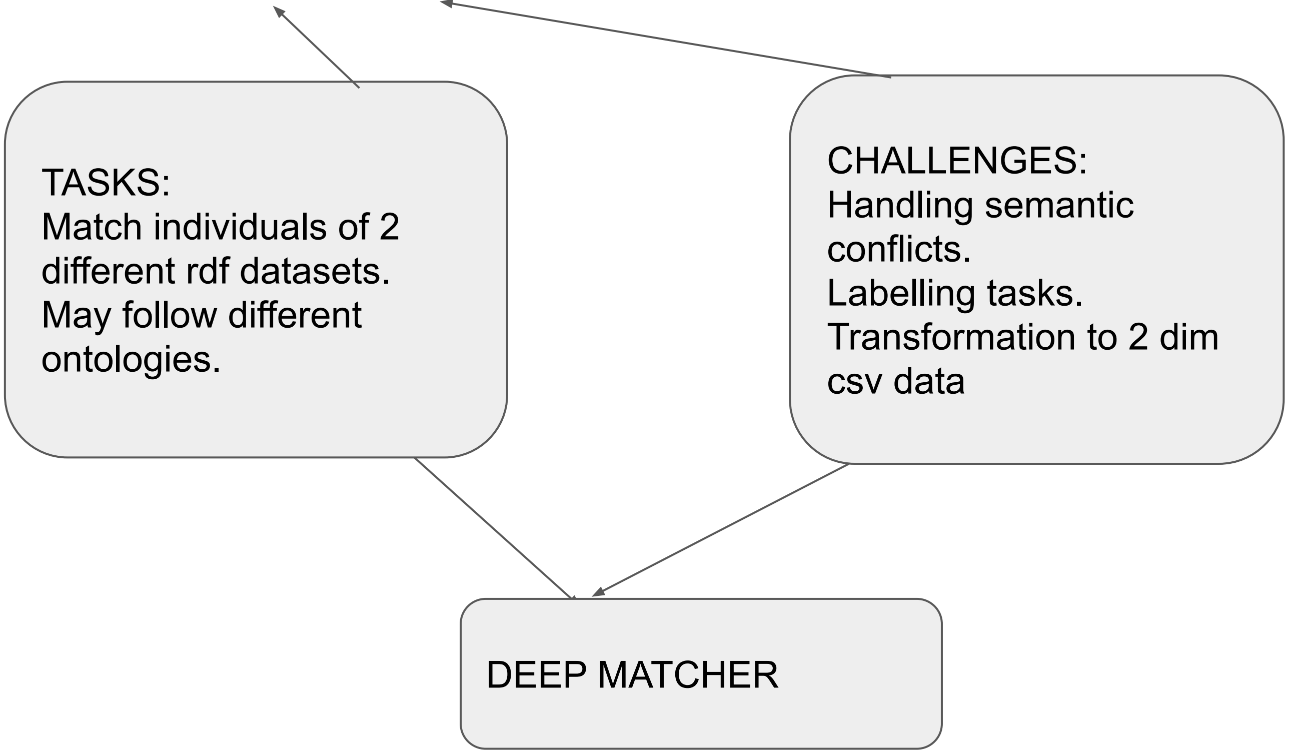


ID	Name	Telephone	Address	Items Purchased
233	Angelica J. Jordan	334-555-0178	111 Spring Ln, Greenville, AL	5556, 7611
452	Angie Jordan	202-555-5477	45 Krakow St, Washington, DC	2297
699	Andrew Jordan	334-555-0178	111 Spring Ln, Greenville, AL	1185, 2299, 3720
720	Angie Jrodon			5556
821	Angelica Jeffries Jordan	202-555-5477	397 Hope Blvd, Greenville, AL	7611

Table above contains shopping data of customers and multiple records can belong to same person

PROBLEM STATEMENT

- Entity matching on RDF data



TASKS:
Match individuals of 2 different rdf datasets.
May follow different ontologies.

CHALLENGES:
Handling semantic conflicts.
Labelling tasks.
Transformation to 2 dim csv data

DEEP MATCHER

THE WHY

SEMANTIC INTEROPERABILITY CONFLICTS

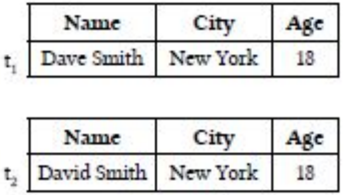
RDF datasets can have different interoperability conflicts

- Variable date formats : dd/mm/yy or mm/dd/yy or dd/yy
- Different level of details : Kgs as grams / euros as cents
- Different formats: centrigrades or fahrenheit/ $ or EUR
- Synonyms or Acronyms : David guetta/ D Guetta
- Different Notation: 2.5 or 2:5

HANDLING THE ABOVE CONFLICTS IS KEY FOR EFFICIENT ENTITY MATCHING!!!

DEEP MATCHER FEATURES

STRUCTURED DATA:



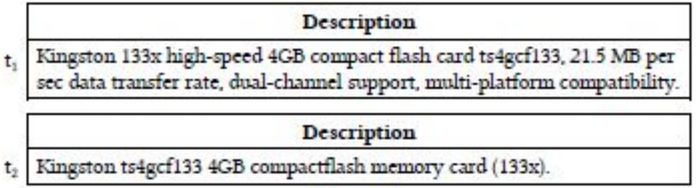
	Name	City	Age
t1	Dave Smith	New York	18

	Name	City	Age
t2	David Smith	New York	18

(a) structured

1. Attribute values are properly aligned
2. Information that is associated only with the attribute
3. Restricted Length

TEXTUAL DATA:



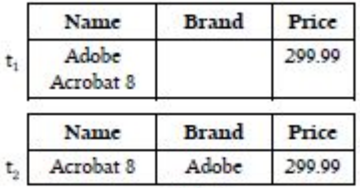
	Description
t1	Kingston 133x high-speed 4GB compact flash card ts4gcf133, 21.5 MB per sec data transfer rate, dual-channel support, multi-platform compatibility.

	Description
t2	Kingston ts4gcf133 4GB compactflash memory card (133x).

(b) textual

All attributes for entity mentions correspond to raw text entries

DIRTY DATA:



	Name	Brand	Price
t1	Adobe Acrobat 8		299.99

	Name	Brand	Price
t2	Acrobat 8	Adobe	299.99

(c) dirty

1. Attribute values may be “injected” under the wrong attribute

ENTITY MATCHING USING DEEP MATCHER

- Cannot handle semantic conflicts
- RDF data cannot be used directly
- 2 dimensional data in specified format is mandatory
- Can handle noisy and raw textual data

DEEP MATCHER WITH CONFLICTS

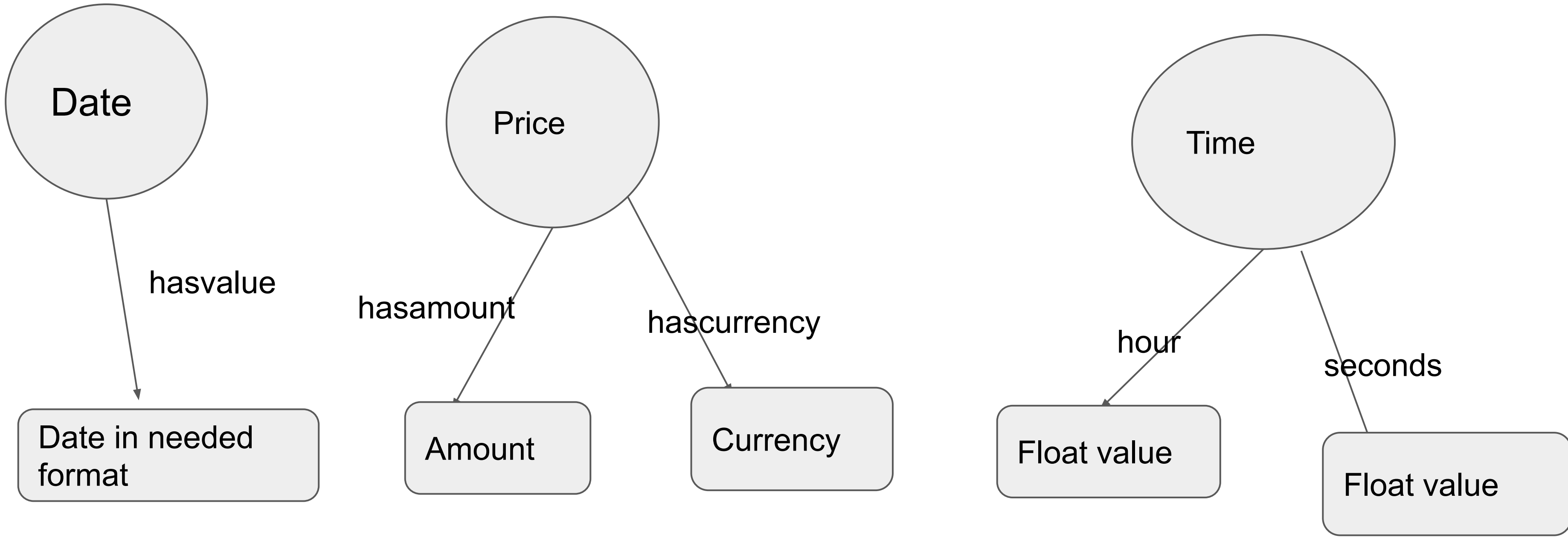
Epochs=10 Dataset= ITunes

Conflicts	F1	Precision	Recall	Method
No	85.25	86.67	83.87	Hybrid
No	86.67	89.66	83.87	RNN
No	75.86	81.48	70.97	Attention
Yes	81.97	83.33	80.65	Hybrid
Yes	83.58	77.78	90.32	RNN
Yes	75.86	81.48	70.97	Attention
Yes	80.00	76.47	83.87	Hybrid

DEEP MATCHER FORMAT

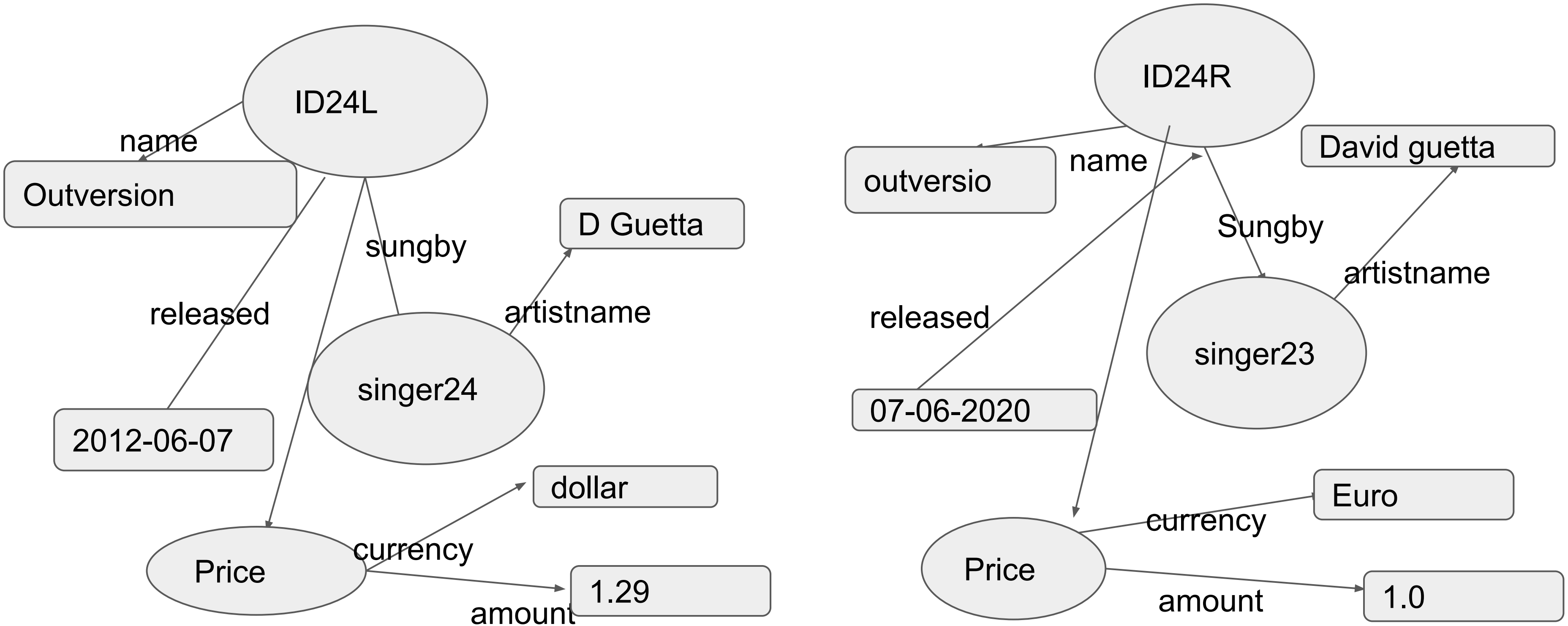
- Left_ prefix for left record columns
- Right_ prefix for right record columns
- Labelling 1 and 0
- ID used like a primary key
- Sorting of columns

RDF REPRESENTATION



RDF representation can help sort conflicts !

PROBLEM IN HAND



MATCH

RDF IN SEMI STRUCTURED DATA ON DEEP MATCHER

OUR CONTRIBUTION

- Entity matching on RDF datasets (Deduce an approach)
- RDF data with conflicts sorted using sparql queries and re designing ontology
- Improves deep matcher accuracy on noisy semi structured data
- Last but not least a transformation algorithm suiting entity matching

THE HOW

ALGORITHM

INPUT

2 similar rdf datasets

Label indicating matched entity across (URI Mapping)

Defined interoperability conflicts

Python environment

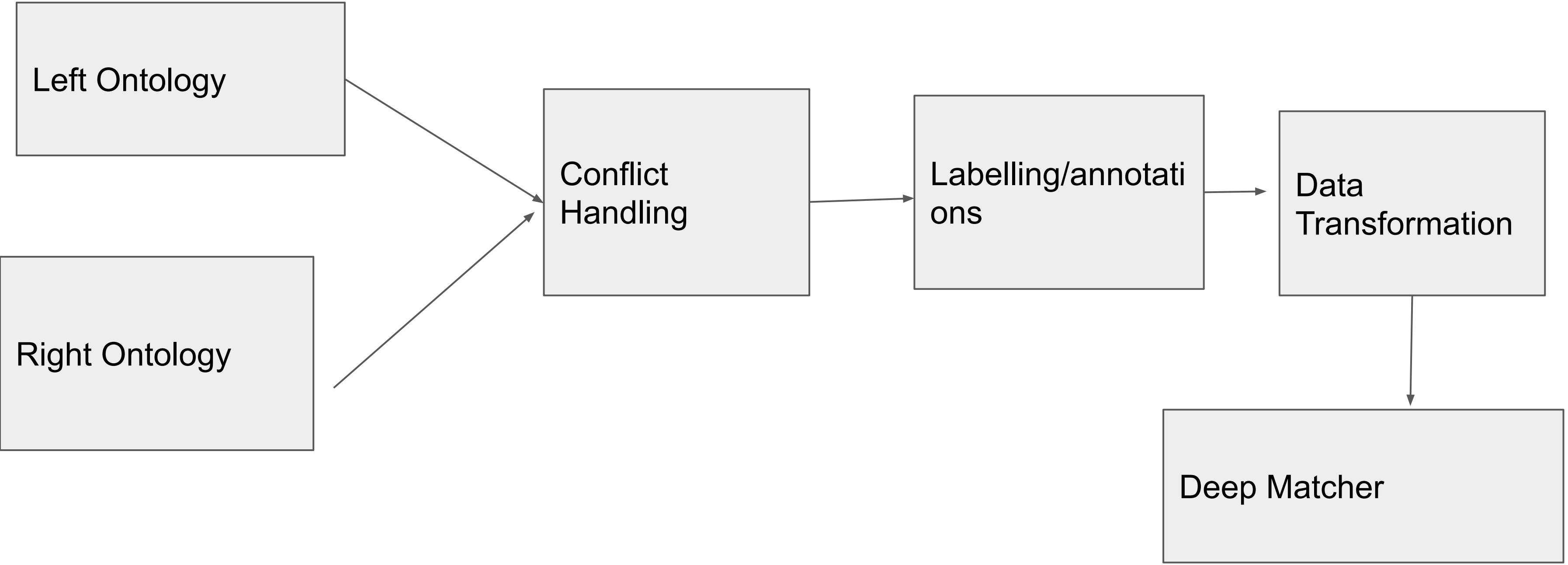
OUTPUT

Transformed rdf data to csv

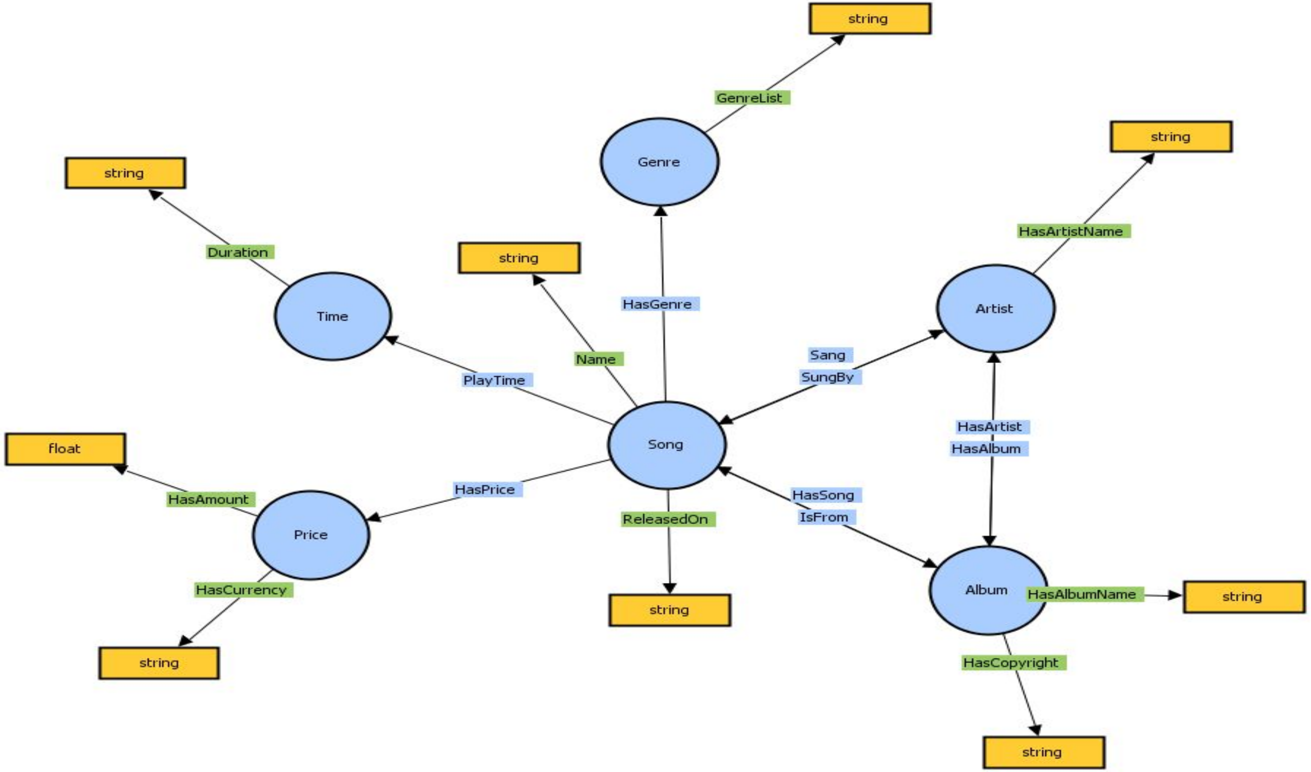
Modified RDF datasets with altered ontology and conflicts better handled

List of matched entities

PIPELINE


Left Ontology
Right Ontology
Conflict Handling
Labelling/annotations
Data Transformation
Deep Matcher

ITUNES ONTOLOGY



HANDLING ITUNES CONFLICTS

Date conflict:

- **Sparql fetch**
- **Regex for identifying conflicts**
- **Fix a format**
- **Sparql update**

Price Conflict:

- **Sparql fetch**
- **Calculate conversion**
- **Change ontology**
- **Sparql update**

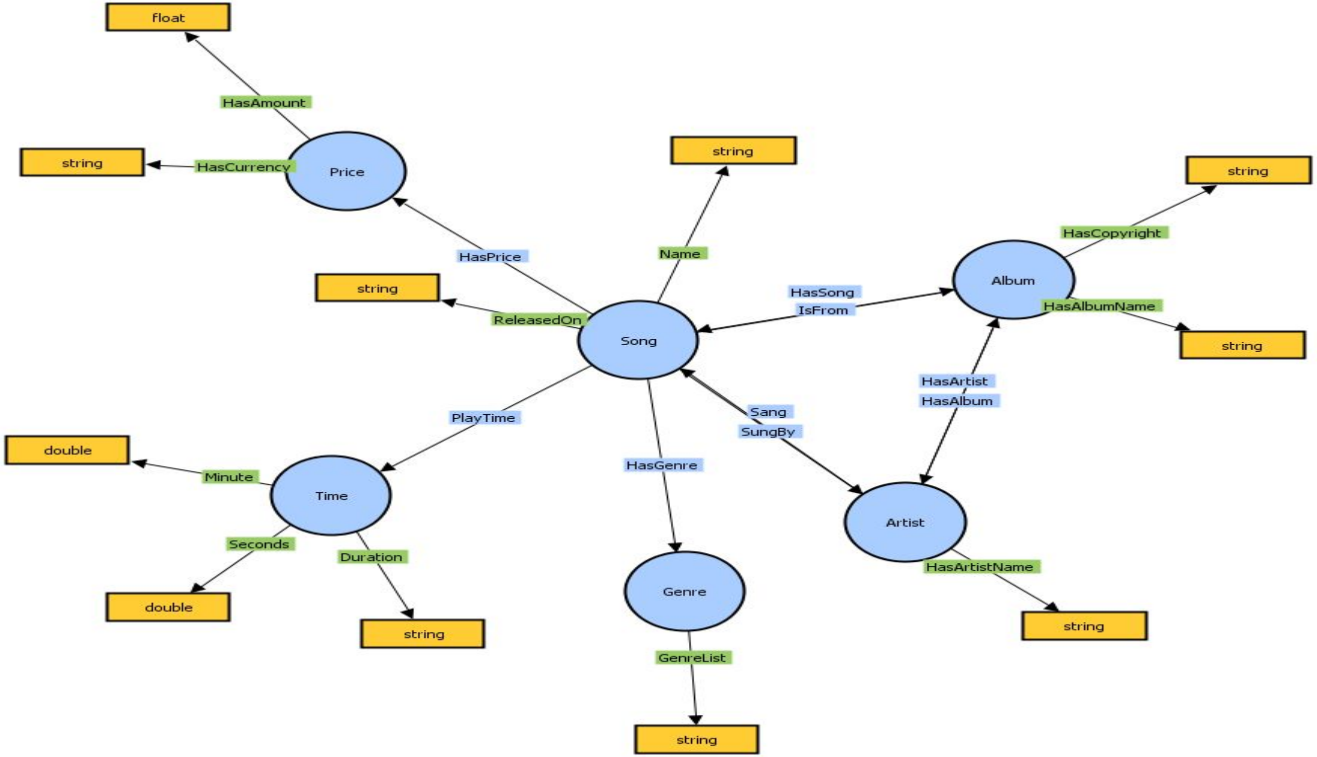
Duration conflict

- **Sparql fetch**
- **Identify all formats by regex**
- **Add entity, change ontology**
- **Sparql update**

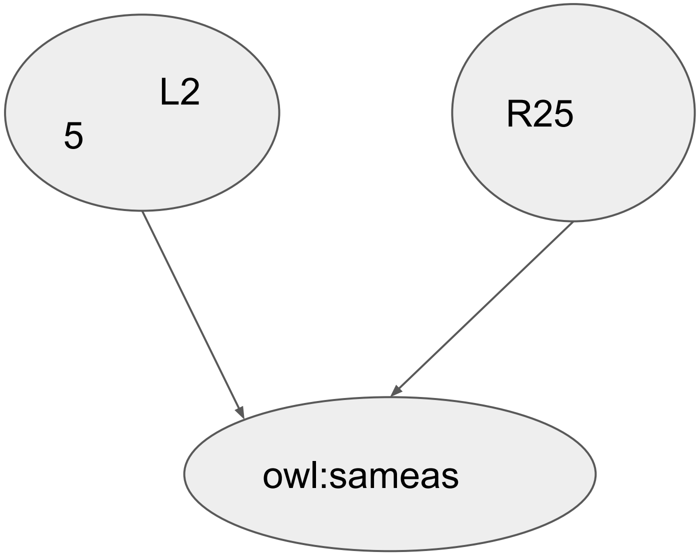
ACRONYM CONFLICT

- **Sparql fetch entity with acronym**
- **Identify entities which depend on acronym entity and use subset feature to see if its the same entity represented with acronym**
- **String match**
- **Update individuals and map all similar individuals into one**
- **Update**

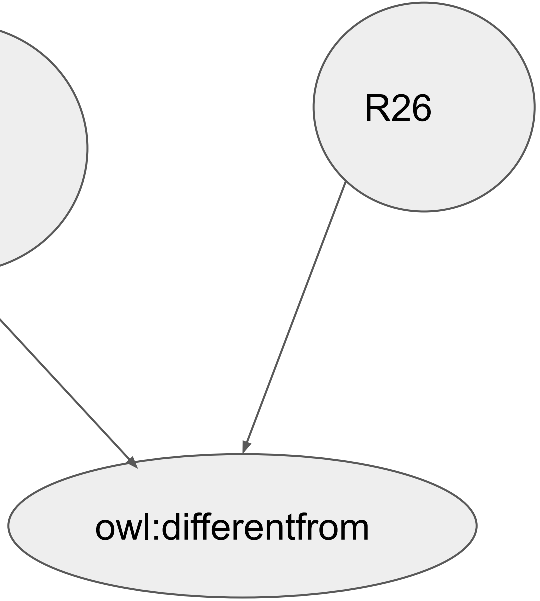
UPDATED ITUNES ONTOLOGY



LABELLING AND ITS TRANSFORMATION

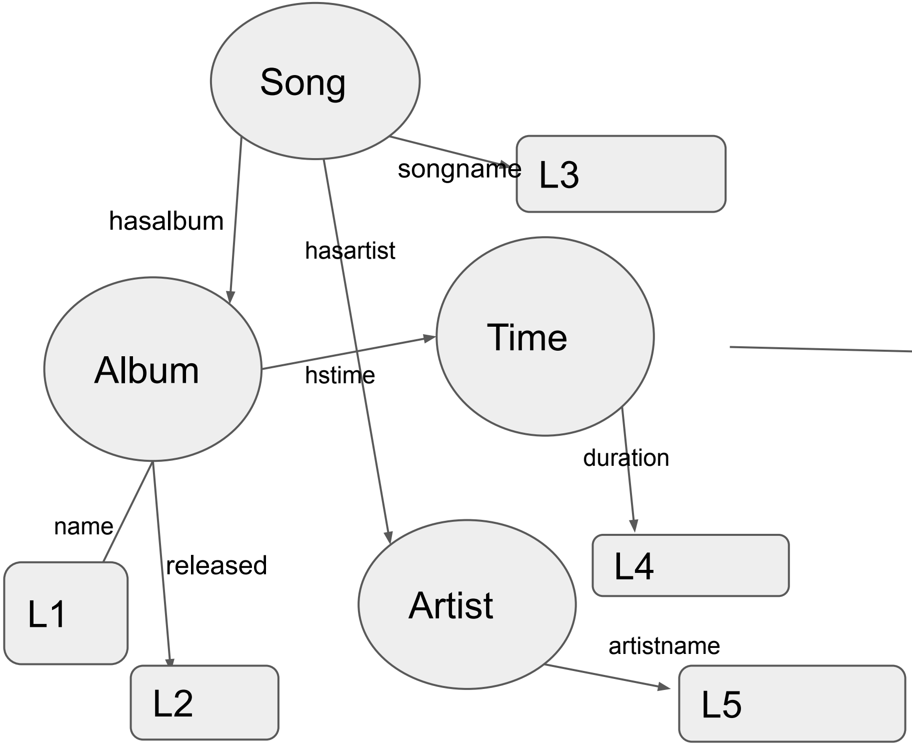


ID	Label
25	1



ID	Label
26	0

DATA TRANSFORMATION



Shortest path using Breadth first search from song (entity to be matched)

L1: (Song,hasalbum,Album), (Album,name,L1)

L2:(Song,hasalbum,Album),(Album,released,L2)

L3:(Song,songname,L3)

L4:(Song,hasalbum,Album),(Album,hastime,Time),(Time,duration,L4)

L1	L2	L3	L4	L5

DEEP MATCHER FOR STRUCTURED DATA

- 2 dim data as i/p
- Semantic interoperability conflicts are sorted
- Semi structured to structured
- Labelling

ID	Label	Left_L1	Left_L2	Left_L3	Right_L1	Right_L2	Right_L3

EXPERIMENTS

KG	Conflcits	F1	Precision	Recall	Method
No	No	85.25	86.67	83.87	Hybrid
No	Yes	79.31	85.19	74.19	Hybrid
Yes	Yes	82.76	88.89	77.42	Hybrid

Thank You