

# Web Data mining and Semantics



Shubham SAINI  
Manisha Priyadarshini RAWLA

# Part I: Modeling the ontology

# Movie Ontology

Active ontology × Entities × Individuals by class × SPARQL Query ×

Classes Object properties Data properties Annotation properties Datatypes Individuals

Class hierarchy: Director

owl:Thing

- Genre
  - Movie
    - Action
    - Adventure
    - Animation
    - Biography
    - Comedy
    - Crime
    - Director
    - Drama
    - Family
    - Fantasy
    - History
    - Horror
    - Music
    - Mystery
    - Program
    - Romance
    - Sci-Fi
    - Sport
    - Thriller
    - War

Person

- Actor
- Director
- Female
- Male
- Writer

- Theater

Asserted

Director — http://www.semanticweb.org/tai/ontologies/2023/2/untitled-ontology-20#Director

Annotations Usage

Annotations: Director

Annotations +

Description: Director

isDirectedby some Director

Instances +

⬢ "EthanCoen,JoelCoen"	? @ ×
⬢ "JoelCoen,EthanCoen"	? @ ×
⬢ "LanaWachowski,LillyWachowski"	? @ ×
⬢ "RogerAllers,RobMinkoff"	? @ ×
⬢ AdrianLyne	? @ ×
⬢ AlexProyas	? @ ×
⬢ AlfredHitchcock	? @ ×
⬢ BarryLevinson	? @ ×
⬢ Bryan Singer	? @ ×
⬢ ChristopherNolan	? @ ×
⬢ DarrenAronofsky	? @ ×
⬢ DavidFincher	? @ ×
⬢ DavidLean	? @ ×
⬢ EmileArdolino	? @ ×
⬢ FrancisFordCoppola	? @ ×
⬢ FrankDarabont	? @ ×
⬢ GeorgeLucas	? @ ×

To use the reasoner click Reasoner > Start reasoner ☐ Show Inferences !

# Object properties

Classes

Object properties

Data properties

Annotation properties

Datatypes

Individuals

Object property hierarchy:

owl:topObjectProperty

hasActor

hasDirected

isDirectedby

isWrittenby

Asserted

# **PART - II**

# **Populating the**

# **Ontology**

CSV\_to\_RDF

File Edit View Insert Runtime Tools Help All changes saved

Files

..

sample\_data

movies.csv

+ Code + Text

✓ 0s [4] # Defining the ontology

ont = Namespace('http://www.semanticweb.org/tai/ontologies/2023/2/untitled-ontology-20#')

movie\_class = ont.Movie

genre\_class = ont.Genre

actor\_class = ont.Actor

director\_class = ont.Director

writer\_class = ont.Writer

✓ 0s [5] graph = Graph()

✓ 0s

import io

with io.open('/content/movies.csv', newline='', encoding= 'latin-1') as csvfile:

reader = csv.DictReader(csvfile)

for row in reader:

# Create a new URI for the movie

movie\_uri = URIRef(ont + row['Movie'].replace(' ', '\_'))

# Add triples for the genres

for genre in row['Genre'].split(', '):

genre\_uri = URIRef(ont + genre.replace(' ', '\_'))

graph.add((genre\_uri, RDF.type, genre\_class))

graph.add((movie\_uri, RDF.type, genre\_uri))

# Add triples for the actors

for actor in row['Actors'].split(', '):

actor\_uri = URIRef(ont + actor.replace(' ', '\_'))

graph.add((movie\_uri, ont.hasActor, actor\_uri))

graph.add((actor\_uri, RDF.type, actor\_class))

# Add triples for the director

for director in row['Director'].split(', '):

director\_uri = URIRef(ont + row['Director'].replace(' ', '\_'))

graph.add((movie\_uri, ont.isDirectedby, director\_uri))

graph.add((director\_uri, RDF.type, director\_class))

# Add triples for the writers

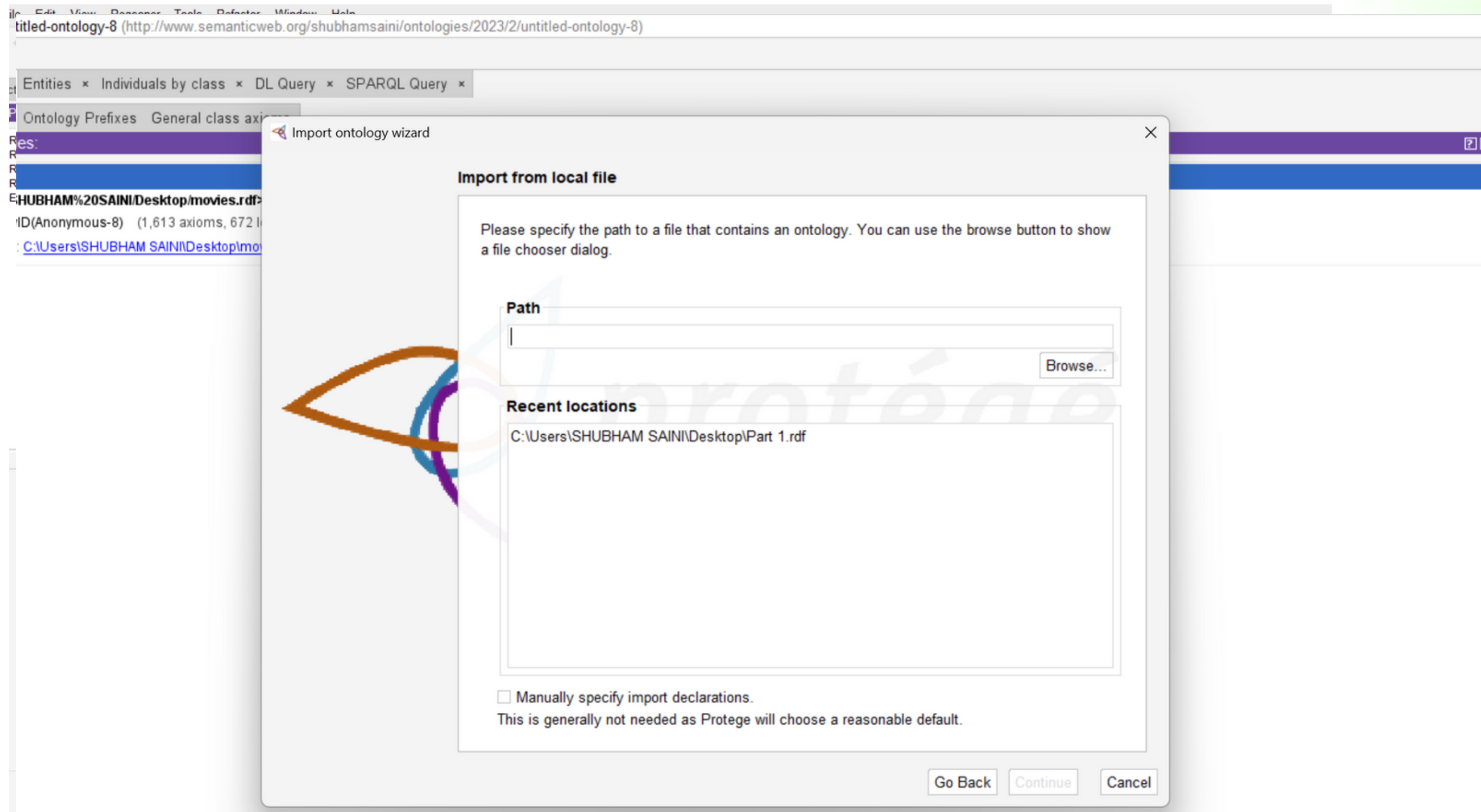
for writer in row['Writer'].split(', '):

RAM

Disk

82.28 GB available

✓ 0s completed at 1:00 PM





## Property Heirarchy

Annotation property hierarchy.




- country
- hasActor
- isDirectedby
- isWrittenby
- language
- owl:backwardCompatibleWith
- owl:deprecated
- owl:incompatibleWith
- owl:priorVersion
- owl:versionInfo
- rdfs:comment
- rdfs:isDefinedBy
- rdfs:label
- rdfs:seeAlso
- title
- year





# Datatypes

Active ontology × Entities × Individuals by class × SPARQL Query ×

Classes Object properties Data properties Annotation properties Datatypes Individuals

Datatypes: 

- owl:rational
- owl:real
- rdf:langString
- rdf:PlainLiteral
- rdf:XMLLiteral
- rdfs:Literal
- xsd:anyURI
- xsd:base64Binary
- xsd:boolean
- xsd:byte
- xsd:dateTime
- xsd:dateTimeStamp
- xsd:decimal
- xsd:double
- xsd:float
- xsd:hexBinary
- xsd:int
- xsd:integer
- xsd:language
- xsd:long
- xsd:Name
- xsd:NCName
- xsd:negativeInteger
- xsd:NMTOKEN
- xsd:nonNegativeInteger
- xsd:nonPositiveInteger
- xsd:normalizedString
- xsd:positiveInteger
- xsd:short
- xsd:string
- xsd:token
- xsd:unsignedByte
- xsd:unsignedInt
- xsd:unsignedLong
- xsd:unsignedShort

<http://www.w3.org/2001/XMLSchema#base64B>

# PART III

# QUERYING THE

# ONTOLOGY

---

# Apache Jena

# Fuseki

Download

<https://jena.apache.org/download/>

## Apache Jena Fuseki

The binary distribution of Fuseki2 (this includes both the standalone and WAR file packaing) may be downloaded at:

[apache-jena-fuseki-2.4.0.tar.gz](#) (MD5, PGP)

[apache-jena-fuseki-2.4.0.zip](#) (MD5, PGP)











Fuseki1, built for the current version of Jena is available at:

[jena-fuseki1-1.4.0-distribution.tar.gz](#) (MD5, PGP)

[jena-fuseki1-1.4.0-distribution.zip](#) (MD5, PGP)

# Starting Fuseki Web Application

apache-jena-fuseki-2.4.0 ➤ apache-jena-fuseki-2.4.0

<input type="checkbox"/> Navn ^	Endringsdato	Type
<input type="checkbox"/>  bin	14.07.2016 14.33	Filmappe
<input checked="" type="checkbox"/>  run	14.07.2016 14.41	Filmappe
<input checked="" type="checkbox"/>  webapp	14.07.2016 14.33	Filmappe
<input checked="" type="checkbox"/>  fuseki	14.07.2016 14.33	Fil
<input checked="" type="checkbox"/>  fuseki.war	14.07.2016 14.33	WAR-fil
<input checked="" type="checkbox"/>  fuseki-server	14.07.2016 14.33	Fil
<input checked="" type="checkbox"/>  fuseki-server.bat	14.07.2016 14.41	Satsvis Windows-fil
<input checked="" type="checkbox"/>  fuseki-server.jar	14.07.2016 14.33	Executable Jar File
<input checked="" type="checkbox"/>  LICENSE	14.07.2016 14.33	Fil
<input checked="" type="checkbox"/>  NOTICE	14.07.2016 14.33	Fil

Apache J...Apache J...Apache J...X

localhost:3030Søk

Apache Jena Fuseki

Apache Jena Fuseki

Version 2.4.0. Uptime: 29m 54s

Datasets on this server

There are no datasets on this server yet. [Add one.](#)

Dataset

Manage datasets

Help

Run queries and modify datasets hosted by this server.

Administer the datasets on this server, including adding datasets, uploading data and performing backups.

Summary of commands and links to online documentation.



```

1 prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 prefix owl: <http://www.w3.org/2002/07/owl#>
4 prefix xsd: <http://www.w3.org/2001/XMLSchema#>
5 prefix : <http://www.uia.no/film_ontology#>
6
7 SELECT ?actress ?movie
8 WHERE {
9     :JamesDean :playedIn ?movie .
10    ?actress :playedIn ?movie .
11    ?actress rdf:type :Woman
12 }
13

```

#### QUERY RESULTS



Table

Raw Response



Showing 1 to 5 of 5 entries

	actress	movie
1	:CarrollBaker	:Giant
2	:ElizabethTaylor	:Giant
3	:MercedesMcCambridge	:Giant
4	:JoVanFleet	:EastOfEde
5	:JulieHarris	:EastOfEde

*Thank  
you!*