

Εργαστήριο Σημασιολογικού Ιστού

Ενότητα 7: Χρήση RDFS και λεξιλογίων RDF

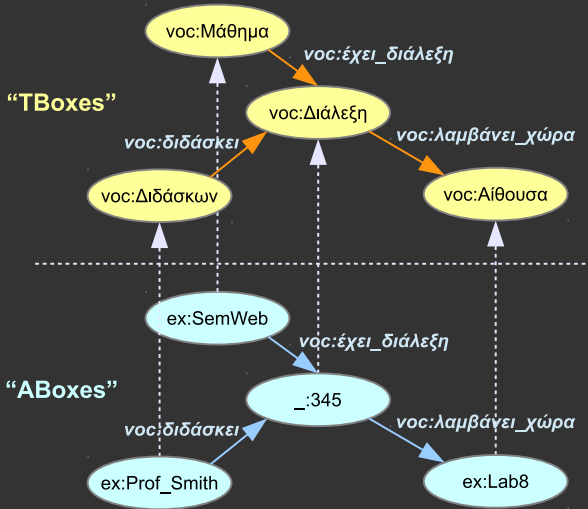
Μ.Στεφανιδάκης

26-4-2018

Επανάληψη: RDFS και λεξιλόγια RDF

- ▶ **RDFS: σημασιολογική επέκταση της RDF**
 - ▶ Το “σχήμα” RDFS βασίζεται στις **κλάσεις** (classes) και τις **ιδιότητες** (properties)
- ▶ **Λεξιλόγια RDF (RDF Vocabularies)**
 - ▶ Παρέχουν όρους (“λέξεις”) για την περιγραφή των κλάσεων και σχέσεων (του “**σχήματος**”, δηλαδή) των RDF δεδομένων μας
 - ▶ όλα είναι URIs σε **κοινό χώρο ονομάτων**
- ▶ Κλειδί για την επιτυχία του Σημασιολογικού Ιστού είναι η **(επανα)χρησιμοποίηση** κοινών λεξιλογίων
 - ▶ Η αξία ενός λεξιλογίου RDF αυξάνεται ανάλογα με τη χρήση του
 - ▶ **Πάντα:** πρέπει να αναζητήσετε υπάρχοντα λεξιλόγια RDF πριν φτιάξετε ένα δικό σας!

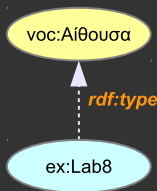
Δύο επίπεδα: Σχήμα και Δεδομένα



TBox (terminology) και **ABox** (assertion): όροι από την επιστήμη της Αναπαράστασης Γνώσης (εδώ σε ανεπίσημη χρήση)

Η σύνδεση Σχήματος και Δεδομένων

- ▶ Το σχήμα (στη μορφή λεξιλογίου RDF) συνήθως βρίσκεται σε ξεχωριστό έγγραφο
 - ▶ Οι τριάδες RDF περιγράφουν οντολογικές έννοιες (TBoxes)
- ▶ Τα “κανονικά” δεδομένα αποτελούν άλλο σετ τριάδων RDF (ABoxes)
 - ▶ Όπου χρησιμοποιούνται τα URIs του λεξιλογίου ως κατηγορήματα ή αντικείμενα
- ▶ Η σύνδεση επιτυγχάνεται μέσω του **rdf:type**



`ex:Lab` **rdf:type** `voc:Αίθουσα` .

Τι μπορεί να κάνει το RDFS για τις εφαρμογές μας;

- ▶ Επίσημα: **τίποτα!**
 - ▶ Δηλαδή: το πρότυπο RDFS **δεν ορίζει** πώς θα χρησιμοποιηθεί από μια εφαρμογή!
 - ▶ Ορίζει μόνο τη σημασιολογία...
- ▶ Οι χρήσεις όμως είναι πολλές και ενδιαφέρουσες:
 - ▶ Εφαρμογές που δρουν με βάση τη σημασιολογία των δεδομένων RDFS
 - ▶ Η ροή ελέγχου καθορίζεται από τα (μέτα)δεδομένα (data-driven applications)
 - ▶ Ανακάλυψη και έλεγχος συνοχής σε σημασιολογικά δεδομένα
 - ▶ Εάν ένα σετ δεδομένων τηρεί ένα “σχήμα”
 - ▶ Εξαγωγή νέας γνώσης πέραν των δηλωμένων τριάδων
 - ▶ Διαδικασία εξαγωγής συμπερασμάτων (inference)

Παράδειγμα εξαγωγής νέας γνώσης

Έστω το σετ δεδομένων RDF:

voc:Course rdf:type rdfs:Class .

voc:UndergraduateCourse rdf:type rdfs:Class .

voc:UndergraduateCourse rdfs:subClassOf voc:Course .

ex:SemWeb rdf:type voc:Course .

ex:CSIntro rdf:type voc:UndergraduateCourse .

Στην ερώτηση: “δώσε τα μαθήματα (voc:Course)”, τα δηλωμένα δεδομένα επιστρέφουν:

- ▶ ex:SemWeb

Αν το πρόγραμμά μας υλοποιεί τους κανόνες RDFS, η απάντηση θα είναι:

- ▶ ex:SemWeb
- ▶ ex:CSIntro

Η σειρά σας!

- ▶ Χρησιμοποιήστε τη βιβλιοθήκη **rdflib** της Python
 - ▶ Παραδείγματα απλής χρήσης στις επόμενες διαφάνειες!
- ▶ Η εφαρμογή σας:
 - ▶ Θα φορτώνει τριάδες από τη διεύθυνση `http://dbpedia.org/data/Lodovico_Giustini.ntriples`
 - ▶ Θα εντοπίζει και θα τυπώνει όλα τα URIs τύπου `foaf:Person` και `foaf:Agent`
 - ▶ Υπάρχουν **ρητά δηλωμένα** `foaf:Agent`;
 - ▶ Στη συνέχεια θα φορτώνει σε ξεχωριστό γράφο το **λεξιλόγιο FOAF** (μορφή xml)
 - ▶ Χρησιμοποιώντας την πληροφορία `rdfs:subClassOf`, μπορείτε να βρείτε URIs που έχουν **έμμεσα** τύπο `foaf:Agent`;

Απλή χρήση rdflib (1)

```
import rdflib

# rdflib knows basic namespaces
from rdflib.namespace import FOAF, RDF, RDFS, XSD

# create an RDF graph object
g = rdflib.Graph()

# load a dataset from a file into graph
g.parse("schedule.nt", format='nt')

# load a dataset from URL
g.parse("http://xmlns.com/foaf/spec/", format='xml')

# get number of triples in graph
print(len(g))

# save graph into a new file
g.serialize("output.ttl", format='turtle')

# iterate through all triples
for s,p,o in g:
    print(s,p,o)

# get a set of all unique predicates
# g.subjects() and g.objects() also exist
preds = set(g.predicates())

# iterate through selected triples
# None here means 'any'
for s,p,o in g.triples((None,RDF.type,None)):
    print(o)
```


Απλή χρήση rdflib (2)

```
# create new URI
semw = rdflib.URIRef('http://ex.com/resource/SemWeb')
print(type(semw)) # rdflib.term.URIRef
print(semw)       # http://ex.com/resource/SemWeb
print(semw.n3())  # <http://ex.com/resource/SemWeb>

# create a URI within a namespace
ns = rdflib.Namespace('http://ex.com/resource/')
semw2 = ns.SemWeb      # same as semw
semw3 = ns['SemWeb']    # same as semw

# create a new blank node
bn = rdflib.BNode()
print(type(bn)) # rdflib.term.BNode
print(bn)       # Nf8cd657ef6cd4a72b0...
print(bn.n3())  # _:Nf8cd657ef6cd4a72b0...

# create a new literal
l = rdflib.Literal('Prof.Smith')
print(type(l)) # rdflib.term.Literal
print(l)       # Prof.Smith
print(l.n3())  # "Prof.Smith"
```

Απλή χρήση rdflib (3)

```
# create typed literal
n = rdflib.Literal('13:00:00.0',datatype=XSD.time)
print(type(n.value))    # datetime.time
# or by passing a python object directly
n2 = rdflib.Literal(7.3)
print(type(n2.value))   # float

# create a literal string with language
la = rdflib.Literal('test',lang='en')
print(la.language)      # en

# add a new triple
g.add((bn,semw,la))
if (bn,semw,la) not in g:
    print('ERROR')

# remove a triple
g.remove((bn,semw,la))
if (bn,semw,la) in g:
    print('ERROR')
```

(κατεβάστε τον **συνολικό κώδικα**)

Η σειρά σας (2)

- ▶ Τροποποιήστε το πρόγραμμά σας, έτσι ώστε να εκμεταλλεύεται και την πληροφορία `rdfs:domain` και `rdfs:range` για την εξαγωγή τύπων.
- ▶ Μπορείτε να ψάξετε για `foaf:Agent` στα δικά σας δεδομένα του ωρολογίου προγράμματος;
 - ▶ Προσθέστε τις κατάλληλες δηλώσεις `rdf:type` στα δεδομένα σας!
 - ▶ Ποιο URI είναι Μάθημα, Αίθουσα, Διδάσκων, Διάλεξη

Μια στιγμή!

- ▶ Μήπως η εφαρμογή σας επιτρέπει μια διάλεξη να γίνεται ταυτόχρονα σε πολλές αίθουσες;
 - ▶ Θα μπορούσε να ξέρει από το λεξιλόγιο RDF ότι δεν είναι σωστό;
 - ▶ Πώς θα εκφράσουμε ότι μια συγκεκριμένη διάλεξη γίνεται **το πολύ σε μία** αίθουσα την ίδια στιγμή;
- ▶ Γενικότερα, αν ένα υποκείμενο μπορεί να συνδεθεί **ακριβώς N φορές** μέσω κατηγορήματος r με διάφορα αντικείμενα;
 - ▶ Το RDFS δεν μπορεί να εκφράσει αυτόν τον περιορισμό!

Επίσης...

- ▶ Πώς θα καθησυχάσετε τους “θεωρητικούς” πως
 - ▶ “Ό,τι είναι Αίθουσα δεν είναι Διδάσκων”
 - ▶ (πρακτικά το γνωρίζουμε, βέβαια)
- ▶ Το RDFS δεν μπορεί να εκφράσει το γεγονός ότι οι κλάσεις Αίθουσες και Διδάσκοντες δεν έχουν κοινά στοιχεία!

Επιπλέον...

- ▶ Αν είχατε κλάσεις LabLecture και TheoreticalLecture
- ▶ Και αντίστοιχα κλάσεις LabRoom και TeachingRoom
- ▶ Πώς θα δηλώνατε σε λεξιλόγιο RDF ότι:
 - ▶ όταν η ιδιότητα takesPlaceAt έχει domain LabLecture, τότε έχει range LabRoom και
 - ▶ όταν η ιδιότητα takesPlaceAt έχει domain TheoreticalLecture, τότε έχει range TeachingRoom
- ▶ Με το RDFS μπορούμε να ορίσουμε μόνο καθολικά το domain και range μιας ιδιότητας, όπως η takesPlaceAt, όχι ανα κλάση
- ▶ Για όλα τα παραπάνω, απαιτούνται πιο σύνθετα λεξιλόγια!