Module 6: Ontologies and Semantic Annotation
“I like the Internet. Really, I do. Any time I need a piece of shareware or I want to find out the weather in Bogota... I'm the first guy to get the modem humming. But as a source of information, it sucks. You got a billion pieces of data, struggling to be heard and seen and downloaded, and anything I want to know seems to get trampled underfoot in the crowd."

German foreign minister Westerwelle visits Ghana.

William Hague and Angelina Jolie visit Eastern DRC.

Blackstone Group LP (BX) agreed to buy 23 industrial properties in southern Virginia and the Washington and Baltimore metropolitan areas from First Potomac Realty Trust (FPRE) for $241.5 million.

- We know the type of named entity but nothing more
- What kind of organization is Blackstone Group LP?
- What is the job of William Hague?
- Where is Eastern DRC, what does DRC stand for?

=> only semantics: choice of annotation type name
=> some knowledge hidden deep in JAPE & Code
Need More Semantics:

- To co-reference DRC with “Democratic Republic of Congo”
- To avoid scattered knowledge in JAPE/Java?
  Cities are locations, cities have zip codes, ...
- To disambiguate: which “Washington” (state / city)?
- To use extracted information to allow for queries like:
  - European politicians who visited an African country?
  - Politicians and actors travelling together?
- To use extracted information to add information to our own Database/Knowledge base:
  - Add information about the buying-agreement to our data about Blackstone Group and First Potomac Realty Trust
  - Connect with trading information or other data we have
Semantic Queries in Google

Paris convention and visitors office - Official website - Paris tourism
en.parisinfo.com/
Paris convention and visitors office diffuses all information to organise your stay or your trip in Paris: hotels and loadings, museums, monuments, going out, ...
Our welcome centres - Paris Map - Transports and ... - Getting around - Book online

Paris - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Paris
Coordinates: 48°51′24″N 2°21′03″E / 48.8567°N 2.3508°E / 48.8567; 2.3508. Paris is the capital and largest city of France. It is situated on the river ...
List of tourist attractions in Paris - History of Paris - Demographics of Paris - Portal

Paris.com - Paris Travel Guide and hotel accommodation
www.paris.com/
Paris.com: Paris. France tourist services offering hotel accommodation, holiday apartments. We guide you to the best Paris city tours and things to do!

News for paris

Paris women finally allowed to wear trousers
BBC News - 21 minutes ago
The French government overturns a 200-year-old ban on women wearing trousers in the capital, Paris, dating from November 1800.

Skirts rule lifted: Centuries-old ban on women wearing trousers in Paris is finally axed
Mirror.co.uk - 3 hours ago
Women in Paris finally allowed to wear trousers
Telegraph.co.uk - 1 day ago

Paris | Travel | The Guardian
www.guardian.co.uk/travel/paris
Latest news and comment on Paris from guardian.co.uk.
• 500 million entities that Google “knows” about

• Used to provide more accurate search results

• Summaries of information about the entity being searched

http://googleblog.blogspot.it/2012/05/introducing-knowledge-graph-things-not.html
Semantic Enrichment

- Textual mentions aren't actually that useful in isolation
  - knowing that something is a “Person" isn't very helpful
  - knowing which Person the mention refers to can be very useful
- Disambiguating mentions against an ontology provides extra context
- This is where **semantic enrichment** comes in
- The end product is a set of textual mentions linked to an ontology, otherwise known as **semantic annotations**
- Annotations on their own can be useful but they can also
  - be used to generate corpus level statistics
  - be used for further ontology population
  - form the basis of summaries
  - be indexed to provide semantic search
Automatic Semantic Enrichment

• Use Text Mining, e.g.
  • Information Extraction – recognise names of people, organisations, locations, dates, references, etc.
  • Term recognition – identify domain-specific terms
• Automatically extend article metadata to improve search quality
• Example: using a customised GATE text mining pipeline to enrich metadata in the Envia environmental science repository

http://www.bl.uk/reshelp/experthelp/science/eventsandprojects/enviatbl/index.html
Preliminary flood risk assessment: prepared to meet the Vale of Glamorgan Council's duties to manage local flood risk under the Flood Risk Regulations (2009)

Citation


Description

Title from PDF cover (viewed on June 27, 2012).

Includes bibliographical references (p. 29-30).

Date

2011

Author(s)

Vale of Glamorgan (Wales). Council

Publisher

Barry: Vale of Glamorgan

Subject

Floods, Risk assessment, Wales, Vale of Glamorgan, Maps, Flood forecasting, Flood control, Planning
Mining medical records

- Medical records contain a large amount of unstructured text
  - letters between hospitals and GPs
  - discharge summaries
- These documents might contain information not recorded elsewhere
  - it turns out doctors don't like forms!
  - often information-specific fields are ignored, with everything put in the free text area
Medical Records at SLAM

• NIHR Biomedical Research Centre at the South London and Maudsley Hospital are using text mining in a number of their studies

• They have developed applications to extract:
  – the results of mental state tests, and the date the test was administered
  – education level (high school, university, etc.)
  – smoking status
  – medication history

• They have even had promising results predicting suicides!
Genome Wide Association Studies (GWAS) aim to investigate genetic variants across the whole genome.

- With enough cases and controls, this allows them to state that a given SNP (Single Nucleotide Polymorphism) is related to a given disease.
- A single study can be very expensive in both time and money to collect the required samples.

Can we reduce the costs by analysing published articles to generate prior probabilities for each SNP?
Can Semantic Annotation Cure Cancer?

- In conjunction with IARC (International Agency for Research on Cancer, part of the WHO) we developed a text analysis approach to mine PubMed.

- We showed retrospectively that our approach would have saved over a year's worth of work and more than 1.5 million Euros.

- We completed a new study which found a new cause for oral cancer.
  - Oral cancer is rare enough that traditional methods would have failed to find enough cases to make the study plausible.
Government Web Archive

- We developed a semantic annotation application to process every crawled page in the archive.
- Entities annotated included; people, companies, locations, government departments, ministerial positions, social documents, dates, money....
- Where possible, annotations were linked to an ontology which
  - was based on DBpedia
  - was extended with UK government-specific concepts
  - included the modelling of the evolution of government
- Annotations were indexed to allow for complex semantic querying of the collection
- An exciting demo coming later, but first the boring stuff you need to know
Why ontologies for semantic search?

- **Semantic annotation**: rather than just annotating the word “Cambridge” as a location, link it to an ontology instance
  - Differentiate between *Cambridge, UK* and *Cambridge, Mass.*

- **Semantic search via reasoning**
  - So we can infer that this document mentions a city in Europe.
  - Ontologies tell us that this particular Cambridge is part of the country called the UK, which is part of the continent Europe.

- **Knowledge source**
  - If I want to annotate *strikes* in baseball reports, the ontology will tell me that a *strike* involves a *batter* who is a *person*
  - In the text “BA went on strike”, using the knowledge that BA is a company and not a person, the IE system can conclude that this is not the kind of strike it is interested in
Example Semantic Search Architecture

Linked data (LOD, GEMET, Ord, Survey) → A → B → C → Knowledge Transformation and Integration → Import & Update → Semantic Repository → Semantic Search UI

EnvIA → Semantic Enrichment → Semantic Indexing & Storage

GATE, MÍMIR, OWLIM
What is Semantic Annotation?

Annotation:

*The process of adding metadata to [parts of] a document.*

Semantic Annotation:

*Annotation process where [parts of] the annotation schema (annotation types, annotation features) are ontological objects.*
Semantic Annotation: Basic Idea

• Link annotations to concepts in a knowledge base.
• The annotated text is a “Mention” of a concept in the KB
• We can use the knowledge associated with Mentions in our IE pipeline
  – e.g. Persons have JobTitles, Cities have zip codes
• We can use the knowledge associated with Mentions for “Semantic Search”
• We can use semantically annotated documents to add new facts to our knowledge base

=> We need some way to represent knowledge
Knowledge Base

Would want to represent knowledge for this domain:

- Westerwelle:
  - has job Foreign minister of Germany → a politician
    - Germany → a country, in Europe
  - Member of the Free Democratic Party
    - Free Democratic Party → a political party
    - Political party → an organization
  ...

- Blackstone Group L.P. → a private equity company
  - has NYSE symbol: BX
  - based in: New York City
    - New York City → a city
    - located in: New York State which is located in USA
  ...

Ontology

A formal way to represent knowledge as:

- Concepts of a domain or a set of domains
  “Agelina Jolie”, “Ghana”
- Relationships between concepts
  “New York City is located in New York State”
- Hierarchies of Concepts and Relationships
  “New York City is a City which is a Location”
- Associated Data
  “Blackstone Group has NYSE symbol BX”
- => most widely used formalism is RDF/OWL
What is an Ontology?

- Set of concepts (instances and classes)
- Relationships between them (is-a, part-of, located-in)
- Multiple inheritance
  - Classes can have more than one parent
  - Instances can have more than one class
- Ontologies are graphs, not trees
• Based on RDF(S) - Resource Description Framework (Schema):
  • Everything is identified by a URI: http://dbpedia.org/page/Paris
  • Everything can be expressed as triples of the form
    Subject Predicate Object:
    :City rdfs:subClassOf :Location .
    :Location a rdfs:Class .
    :BlackstoneGroup :hasNyseSymbol “BX” .
  • Simple vocabulary to express things:
    rdf:type = “belongs to a class”
    rdf:Class = “the class of all classes”
    “BX” = the literal string “BX”
• All resources identified by URIs
  Different URIs may refer to the same resource

• Resources that are “Individuals” can be grouped into
  “Classes” and relate to other things and to values by
  “Properties”.

• Values represented through “Literals”:
  “BX” - a string (untyped literal)
  “New York State”@en – string with language tag (untyped)
  “Guido Westerwelle”^^xsd:string – typed literal
  “24”^^xsd:integer

• :A rdf:type :B – :A is contained in class :B
  :B rdf:type rdfs:Class – :B is an RDFS Class
  :B rdfs:subClassOf :C – all members of :B are in :C
OWL Ontologies

- **OWL**: Web Ontology Language
- **Classes/Concepts and Individuals/Instances**
- **Properties**:
  - DatatypeProperty: individual → literal
  - ObjectProperty: individual → individual
  - AnnotationProperty: resource → literal, but no inference
- **Inference/Reasoning**:
  - Inheritance/Subsumption (classes and properties)
  - “Restrictions”: domain, range, allValuesFrom, hasValue ... infer class membership, property values
    - Open World Assumption: what isn’t asserted, we don’t know
    - Non Unique Name Assumption: different names may be used for same entity
- **Classes can have more than one parent, Individuals can belong to more than one class → OWL Ontologies are graphs, not trees**
DBpedia

- Machine readable knowledge on various entities and topics, including:
  - 410,000 places/locations,
  - 310,000 persons
  - 140,000 organisations

- For each entity we have:
  - entity name variants (e.g. IBM, Int. Business Machines)
  - a textual abstract
  - reference(s) to corresponding Wikipedia page(s)
  - entity-specific properties (e.g. latitude and longitude for places)
Example from DBpedia

The Thames Barrier is the world's second-largest movable flood barrier and is located downstream of central London, United Kingdom. Its purpose is to prevent London from being flooded by exceptionally high tides and storm surges moving up from the sea. It needs to be raised (closed) only during high tide; at ebb tide it can be lowered to release the water that backs up behind it.

Links to GeoNames And Freebase
Latitude & Longitude
GeoNames

- 2.8 million populated places
  - 5.5 million alternate names
- Knowledge about NUTS country sub-divisions
  - use for enrichment of recognised locations with the implied higher-level country sub-divisions
- However, the sheer size of GeoNames creates a lot of ambiguity during semantic enrichment
- We use it as an additional knowledge source, but not as a primary source (DBpedia)
Ontologies in GATE

- Can use OWL-Lite ontologies as language resources (→ Plugin Ontology)
- Ontology Editor, Ontology Annotation Tool, Relation Annotation Tool (→ Plugin Ontology_Tools)
- Ontology-enabled JAPE, JAPE Plus
- LKB Gazetteer (→ Plugin Gazetteeer_LKB) OntoRoot Gazetteer (→ Plugin Gazetteeer_Ontology_Based)
- Ontology-based evaluation (→ Plugin Ontology_BDM_Computation)
- Java API for ontology manipulation, triple manipulation, SPARQL queries
GATE Ontology Implementation

- Based on Sesame and the OWLIM-Lite SAIL (Storage and Inference Layer) implementation from Ontotext
- Fast in memory repository, scales to millions of statements (depending on RAM)
- In addition to local file ontology, can connect to server:
  - OWLIM Lite
  - OWLIM SE/Enterprise: commercial product, persistent and scalable implementation for huge (billion triples) ontologies
- Java API represents OWL concepts (ontology, property, literal) as Java objects
- Also provides support for SPARQL and manipulating Triples directly
Load Ontology

- Need plugin Ontology
- For Editor, also need plugin Ontology_Tools
- Language Resource → New → OWLIM Ontology

![Parameters for the new OWLIM Ontology]

- Loaded:
• Basic viewing of ontologies
• Some edit functionalities:
  • create new concepts and instances
  • define new properties and property values
  • deletion
• Some limitations of what's supported, basically chosen from practical needs for semantic annotation
• Not a Protégé replacement
Ontology Editor
Ontology-based IE

John lives in London. He works there for Polar Bear Design.
John lives in London. He works there for Polar Bear Design.
Greece v Argentina: Who wins on penalties?
By Robert Plummer Business reporter, BBC News
Anyone examining the precedents for the Greek financial crisis might well be amused by the draw for next month's football World Cup matches.
Greece's players celebrated after qualifying for the 2010 World Cup.

For, as fate would have it, Greece's foes in Group B include the country that last suffered a comparable economic fiasco: Argentina.

In the worst-case scenario, Argentina's recent past is Greece's future.

The peso collapse, massive default and subsequent social and political unrest that rocked Argentina in 2001-2002 are being seen by many economists as an awful warning for the politicians in Athens and Brussels.

As far as football is concerned, the two teams will play their first and final group match.

But the day of decision for the Group B nations comes on June 12 when Argentina stave off default by honouring bonds.

The EU and the IMF have agreed

<table>
<thead>
<tr>
<th>Type</th>
<th>Set</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>1222</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>1222</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>1222</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>1222</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>1222</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>1222</td>
<td>1228</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>1233</td>
<td>1241</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>1556</td>
<td>1558</td>
<td></td>
</tr>
</tbody>
</table>

Open Search & Annotate tool

[Open Search & Annotate tool](http://dbpedia.org/ontology/Place)
[Open Search & Annotate tool](http://dbpedia.org/resource/Brussels)
[locType] other

[Open Search & Annotate tool](http://dbpedia.org/ontology/Place)

LKB_Location
Typical Semantic Annotation pipeline

- Analyse document structure
- Linguistic Pre-processing
  - NE recognition
- Ontology Lookup
- Ontology-based IE
- Populate ontology (optional)
- Export as RDF
Semantic Annotation with other tools: OpenCalais

http://viewer.opencalais.com/

Paste text of http://www.membranes.com/

Not easily customised/extended

Domain-specific coverage varies
• Paste text from www.membranes.com

• The main entity of interest “Hydranautics” is missed

• Common problem with general purpose, open-domain semantic annotation tools

• Best results require bespoke customisation
Ontology Learning / Population

- Ontology Population: add new facts to a given ontology. The ontology structure and many classes and individuals are already there:
  “Westerwelle visits Ghana”
  \[ \text{:GWestervelle01} :\text{actorOf} :\text{Event001} . \]
  \[ :\text{Event001} a :\text{VisitingEvent} . \]
  \[ :\text{Event001} :\text{destination} :\text{Ghana} . \]
  ...

- Ontology Learning: also create or extend the structure of the ontology.
Semantic Annotation: How

- Manually
  GATE: ontology based annotation using OAT/RAT or through crowdsourcing

- Automatically
  - Gazetteer/rule/pattern based
    GATE: OntoRoot gazetteer, LKB gazetteer, JAPE, ...
  - Classifier (ML) based – see the YODIE lecture later
  - Combination of the two
GATE: Automatic Semantic Annotation

• Ontology aware Gazetteers:
  • LKB Gazetteer
  • Other gazetteers, using inst/class features
• Ontology aware JAPE
• Semantic Enrichment: LKB Gazetteer, JAPE
LKB Gazetteer

- The LKB gazetteer is used to do ontology-based gazetteer lookup against very large ontologies, e.g. DBPedia, GeoNames and other Open Linked Data ontologies
- Uses a SPARQL query to create a gazetteer list from the ontology

```
SELECT DISTINCT ?label ?inst ?class
WHERE {
  FILTER (lang(?label) = "en")
}
```

- Internally retrieves the result rows and converts them to gazetteer entries with inst and class features
- Creates a cache file that will load fast subsequently
LKB: Continued

- Lives in plugin Gazetteer_LKB

- LKB does not use the GATE ontology language resources. Instead, it uses its own mechanism to load and process ontologies.

- Set up your dictionary first. The dictionary is a folder with some configuration files. Use the samples at GATE_HOME/plugins/Gazetteer_LKB/samples as a guide or download a pre-built dictionary from ontotext.com/kim/lkb_gazetteer/dictionaries.

- The dictionary directory defines which repository to connect to, which SPARQL queries to use to initialise the gazetteer, etc.

- For details see http://gate.ac.uk/userguide/sec:gazetteers:lkb-gazetteer
LKB: Example

- Samples in gate/plugins/Gazetteer_LKB/samples/dictionary_from_remote_repository
- An ontology-based gazetteer of actors from DBpedia

```
Query:

1. SELECT ?Name ?Person ?Cls
2. FROM <http://www.ontotext.com/disable-sameAs>
3. WHERE {
5.   FILTER (lang(?Name) = "en")
6.   FILTER (?Cls = <http://dbpedia.org/ontology/Actor>)
7. }
```

- Test this query against http://ldsre.ontotext.com/sparql
- Or just try some of the sample queries there
Results for **PREFIX rdfs:**... (100 of 850)

<table>
<thead>
<tr>
<th>Name</th>
<th>Person</th>
<th>Cls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jet Li@en</td>
<td>dbpedia:Jet_Li</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Tom Cruise@en</td>
<td>dbpedia:Tom_Cruise</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Cruise, Tom@en</td>
<td>dbpedia:Tom_Cruise</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Bruce Lee@en</td>
<td>dbpedia:Bruce_Lee</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Lee Armstrong@en</td>
<td>dbpedia:Lee_Armstrong</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Johnny Depp@en</td>
<td>dbpedia:Johnny_Depp</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Depp, Johnny@en</td>
<td>dbpedia:Johnny_Depp</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Zhang Ziyi@en</td>
<td>dbpedia:Zhang_Ziyi</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Chow Yun-fat@en</td>
<td>dbpedia:Chow_Yun-fat</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Tsui Hark@en</td>
<td>dbpedia:Tsui_Hark</td>
<td>dbp-ont:Actor</td>
</tr>
<tr>
<td>Sammo Hung@en</td>
<td>dbpedia:Sammo_Hung</td>
<td>dbp-ont:Actor</td>
</tr>
</tbody>
</table>
Ontology Aware JAPE

- JAPE transducers have a run-time parameter which is an ontology.
- [Note that the ANNIE NE Transducer] does not have this parameter, so you cannot use it for ontology-aware JAPE.
- By default it is left blank, so not used during LHS matching.
- When an ontology is provided, the class feature can be used on the LHS of a JAPE rule.
- When matching the class value, the ontology is checked for subsumption: any subclass on the left side of “==” matches.
- e.g. \{Lookup.class == Person\} will match a Lookup annotation with class feature, whose value is either Person or any subclass of it.
Ontology-aware JAPE example

Phase: OntoMatching
Input: Lookup
Options: control = appell

Rule: PersonLookup

```
( {Lookup.class == Person}
  :person
  -->
  :person.Mention = 
    {class = :person.Lookup.class,
     inst = :person.Lookup.inst}
```

- Matches the class Person or any of its subclasses
- Adds class and instance information as features on the Mention annotation
Ontology-aware JAPE example

Ontology-aware JAPE applies only to a feature named “class” and only if the PR's ontology parameter is set.

{Lookup.class == “http://example.com/stuff#Person”}

Matches this class or any subclass in the ontology

{Lookup.class == “Person”}

If the string is not a full URI, JAPE adds the default namespace from the ontology, looks up that class in the ontology, and matches it or any subclasses. Be very careful if your ontology uses more than one namespace!

These rules apply equally to the string in the JAPE rule and in the value of the annotation's class feature.
Templates to simplify namespaces

Template declarations can be used to simplify namespaces.

Template: protont =
   “http://proton.semanticweb.org/2005/04/protont#${n}”
...
{Lookup.class == [protont n=Person]}
...
{Lookup.class == [protont n=Location]}

If you switch to a newer version of PROTON, you only need to change the Template declarations, not every JAPE LHS. (See the GATE User Guide http://gate.ac.uk/userguide/sec:jape:templates for more details and examples.)

Template: protont =
   “http://proton.semanticweb.org/2006/05/protont#${n}”
...
Matching subclasses

David Cameron was the first of the main UK party leaders...

The rule matches because Leader is a subclass of Person
Semantic Enrichment

- Add additional knowledge to semantically annotated mentions
- Simplest: add features
e.g. add the name of the country, zip code for a city
  → if we have city names to disambiguate, may use zip code to disambiguate!
- Use Java API in JAPE RHS, Groovy or own PR
- SemanticEnrichment PR from the Gazetteer_LKB plugin
  - SPARQL Endpoint (not GATE Ontology LR)
  - Run SPARQL query for each URI in inst
  - add query result to 'connections' feature
Semantic Enrichment PR

- Adding new data to semantic annotations by querying external RDF (Linked Data) repositories
- A semantic annotation is an annotation that is linked to an RDF entity by having the URI of the entity in the ‘inst’ feature of the annotation
- This PR runs a SPARQL query against a given repository and puts a comma-separated list of the values mentioned in the query output in the ‘connections’ feature of the annotation
- Run-time parameters:
  - List of annotation types to enrich and input AS
  - Delete on no relations (true/false)
  - Query
QUESTIONS?
Extra exercises
LKB: Try it

- **Samples in** gate/plugins/Gazetteer\_LKB/samples/dictionary\_from\_remote\_repository
- Load the ready-made application **sample\_linked\_data\_mashup.gapp**
- This should load the Movie stars pipeline application
- Temporarily move away the LDSR Enrichment PR from the pipeline, leaving just the documents reset and the entertainers gazetteer
  - that’s pre-built from the SPARQL query shown on the previous page
- Run the pipeline on the sample corpus and inspect the Lookup annotations
Hands On: Semantic Enrichment

- Add the LDSR Enrichment PR back into your pipeline, making sure it is last.
- Run the pipeline on the sample corpus and inspect again the Lookup annotations, especially their **connections** feature.
- **You will need internet connection for this to work.**

- How do results change, if you modify the query to say LIMIT 1, instead of LIMIT 10?
Modelling social media with ontologies

• SIOC and SIOC Types Ontologies

• SIOC (Semantically-Interlinked Online Communities) Core Ontology provides concepts and properties, describing information from online communities (e.g. wikis, weblogs)
  – Ontology namespace: http://rdfs.org/sioc/ns#

• SIOC Types adds extensions for Twitter modelling
  – Ontology namespace: http://rdfs.org/sioc/types#

• Open the SIOC Types ontology in GATE (in hands-on), by giving the URL as an RDF/XML parameter to the OWLIM Ontology LR

• Double click to view the ontology
MicroblogPost and some properties
Users modelled through the http://rdfs.org/sioc/ns#UserAccount class

Useful properties for modelling tweet user info:
- sioc:description: corresponds to the description JSON entry
- sioc:name, sioc:email, sioc:id

Properties for relating users to users: follows

Properties for relating users to tweets: creator_of(UserAccount, Post/MicroblogPost)

Modelling tweets: http://rdfs.org/sioc/types# MicroblogPost
- sioc:content, sioc:embeds_knowledge, sioc:has_creator, sioc:has_reply, sioc:links_to, sioc:topic
A word of warning:

- **Watch out for the namespaces!**
- Some are from SIOC, others – SIOCT, and yet others from other imported ontologies, like SKOS
- E.g. http://rdfs.org/sioc/ns#UserAccount
- Vs http://rdfs.org/sioc/types# MicroblogPost
- In JAPE rules, you need to:
  - Either specify the complete URIs, including the namespaces (unless it is the sioct, which is the default name space for this ontology)
  - Or use templates to shorten the NS URIs and make the JAPEs more readable