
Module 9: Ontologies and Semantic Annotation

Ontology – A Definition

- “An Ontology is a formal specification of a shared conceptualisation.” [Gruber]

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

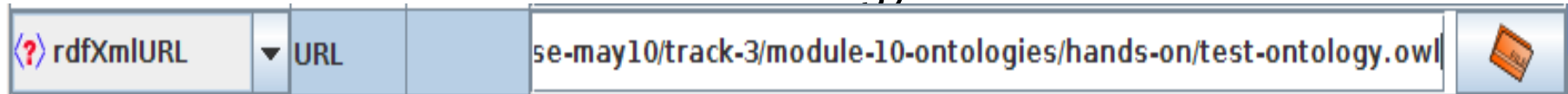


Why ontologies in GATE?

- **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

Ontologies in GATE

- Abstract ontology model for the API, based on the OWL formalism
- Comes with one concrete implementation pre-installed: Sesame/OWLIM
 - Load the Ontology plugin, not OWLIM2 (which is there for backwards compatibility)
 - Instantiate the OWLIM Ontology LR



- GATE provides also several associated tools:
 - Ontology Visualizer/Editor
 - OntoRootGazetteer
 - Ontology support in JAPE

Ontology implementation

- SwiftOWLIM3 from Ontotext
- Fast in memory repository, scales to millions of statements (depending on RAM)
- SwiftOWLIM is exchangeable with persistence-based BigOWLIM: not free, scales to billions of statements
- Supports “almost OWL-Lite”

Ontology Viewer/Editor

- Basic viewing of ontologies, to allow their linking to texts via semantic annotation
- 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 Protege replacement



Ontology Editor

File

Options

Tools

Help

ATE

Applications

Language Resource

protonust-popul...

Processing Resource

Data stores

Messages

protonust-popul...

Classes & Instances

Properties

Classes and Instances

Forest

Jungle

Oasis

Tundra

Cape

Cave

Coast

Continent

Africa

Asia

Europe

North America

South America

Crate

Delta

Gap

Island

Archipelago

Isthmus

Mountain

MountainRange

MountainSummit

Ridge

Volcano

Peninsula

Plain

Plateau

Reef

Valley

Resource Information

Asia

URI

http://gate.ac.uk/owlim#Asia

TYPE

Ontology Instance

Direct Types

Continent

All Types

LandRegion

Location

Continent

Same Instances

hasChild

hasSpouse

hasUniversity

hasMobilePhone

hasFather

hasInternetAddress

hasSister

comment

hasParent

hasContactInfo

hasEMail

More >

[ALL CLASSES]

[ALL CLASSES]

[University]

[ALL CLASSES]

[ALL CLASSES]

[PhoneNumber]

[WebPage]

[ALL RESOURCES]

[ALL RESOURCES]

[Man]

[Man]

[InternetAddress]

[Woman]

[ALL RESOURCES]

[ALL CLASSES]

GATE Ontology Editor

Initialisation Parameters

URIs, Labels, Comments

- The names of a classes or instance shown in the editor is a URI (Universal Resource Identifier)
 - `http://gate.ac.uk/example#Person`
 - URIs cannot have spaces: use underscores instead
- The linguistic lexicalisation is typically encoded in the **label** property, as a string
 - To add a label, right click on the class/instance, select Properties/Label and enter the value in the dialogue box
- The **comment** property is often used for documentation purposes, similarly a string
- Comments and labels are **annotation properties**



New label

_00014

s

Classes and Instances

Entity

Location

City

Sheffield

Organization

A_Company

Person

Diana_Maynard

PERSON

PERSON

All Types

Person

Entity

Same Instances

Property Types

versionInfo

[ALL RESOURCES]

comment

[ALL RESOURCES]

seeAlso

[ALL RESOURCES]

label

[ALL RESOURCES]

person_works_for

[Organization]

isDefinedBy

[ALL RESOURCES]

person_has_age

http://www.w3.org/2001/XMLSchema

Property Values

label

Diana Maynard

lexicalisation

n

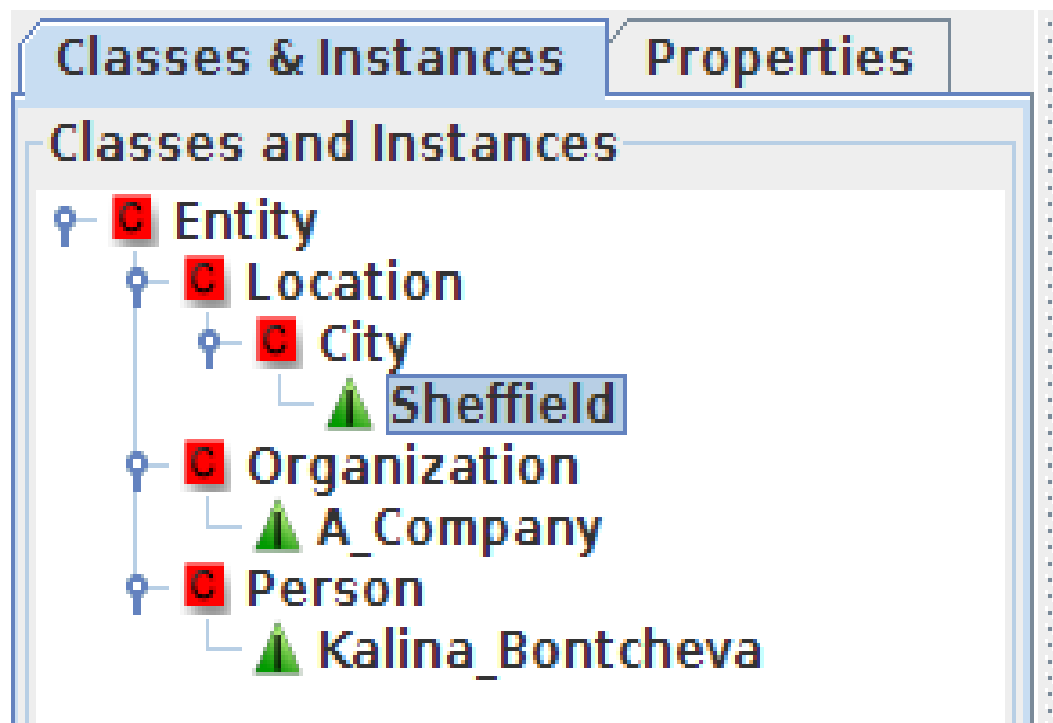
GATE Ontology Editor

Initialisation Parameters

Hands-on 1: classes and instances

- Load the Ontology and Ontology_Tools plugins
 - Language Resource → New → OWLIM Ontology
 - For RdfXmlURL point to **test-ontology.owl**
- This loads the simple ontology of Entity, Location, etc.
 - Double-click on the ontology LR to see it
 - Create a subclass of Location called “City” and then add the city where you live as an instance of City
 - Add yourself as an instance of the class Person
 - Add a label with your full name
- Save the ontology (right click on ontology in resources pane and select “Save as”)
- Keep the ontology open for the next hands on

My ontology



Datatype Properties

- Datatype properties link individuals to data values
- Datatype properties can be of type boolean, date, int,
 - e.g. a person can have an age property
- Available datatypes taken from XMLSchema
- To define a new data property
 - Select an ontology class and click on the D button
 - Choose the desired datatype from the list (e.g. int)
 - Provide the property name (e.g. hasAge)
 - Specify the domain (i.e. the instances of which class it applies to)
 - If more than one class is listed as a domain, these restrict the property to those individuals that belong to the intersection of the class descriptions



Adding a new property

Messages

test-ontology.o...

TC SC R A D O S T X

Classes & Instances

Properties

Classes and Instances

Entity

- Location
- Organization
 - A_Company
- University
 - University_of_Sheffield
- Person
 - Diana_Maynard

Domain

http://gate.ac.uk/example#Person

Add Remove

http://gate.ac.uk/example#Person

OK Cancel

New Datatype Property

Name Space: http://gate.ac.uk/example#

Data Type: http://www.w3.org/2001/XMLSchema#positiveInteger

Property Name: hasAge

Domain

OK Cancel

Label

My Person class

Instances

Diana_Maynard

Diana_Maynard

✗

Adding a property value

- To add a value for an instance, right click on the instance and select “Properties” and then the name of the property for which you want to add a value.
- If the property is not listed, then you haven't defined it yet for the concept to which your instance belongs
- Enter the value in the popup box
- You should now see the property and its value listed in the right hand pane



Adding a property value

Messages test ontology

Classes & Instances Properties

Classes and Instances

- Entity
 - Location
 - City
 - Sheffield
 - Organization
 - A_Company
 - Person
 - Kalina_Bontcheva

1. Select instance and property

2. Add or select value

New property value

Enter a value for the property hasAge of type positiveInteger

12

OK Cancel

▼ Resource Information

- Kalina_Bontcheva
 - URI: http://gate.ac.uk/example#Kalina_Bontcheva
 - TYPE: Ontology Instance
- ▼ Direct Types
 - Person: Person
- ▼ All Types
 - Entity: Entity
 - Person: Person

▼ Properties

- seeAlso
- versionInfo
- comment
- label
- isDefinedBy
- hasAge

▼ Property Values

- hasAge: 12

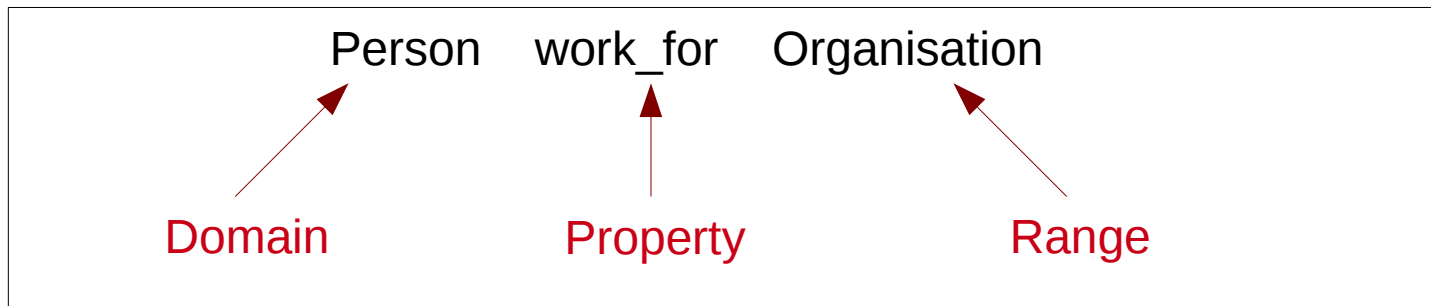
3. Property and value displayed

Hands-on 2: Datatype properties

- Use the entity ontology from the previous exercise
- Add a datatype property “hasAge” with Domain “Person”
 - Add a value for the hasAge property to the instance of Person that refers to you (you can make it up if you don't want to reveal your real age!)
- Add an instance of Organization denoting the organization you work for (make one up if you like)
- Save the ontology (with the same name as before)
- Keep everything open for the next hands-on

Object Properties

- Object properties link instances together
 - They describe relationships between instances, e.g. people work for organisations
- Domain is the subject of the relation (the thing it applies to)
- Range is the object of the relation (the possible "values")



Similar to domains, multiple alternative ranges can be specified by using a class description of the owl:unionOf, but this raises the complexity of the ontology and makes reasoning harder

Creating new Object Properties

- To define a new object property
 - Click on the O button
 - Provide a property name and the values for domain and range
- To set the value of an object property for an instance:
 - Right-click on the instance
 - Select Properties and then the name of the relevant property
 - From the drop down list of instances, choose the correct instance as a value



New Object Property

TC SC R A D O S T X

Classes & Instances

Properties

Classes and Instances

Entity

Location

Organization

A_Company

Person

A_Person

▼ Resource Information

Person

URI

TYPE

http://gate.ac.uk/example#Person

Ontology Class

▼ Direct Super Classes

Entity

Entity

▼ All Super Classes

Entity

Entity

New Object Property

Name Space:

http://gate.ac.uk/example#

Property Name:

person_works_for

Domain

Range

OK

Cancel

Domain

http://gate.ac.uk/example#Entity

Add

Remove

http://gate.ac.uk/example#Person

OK

Cancel

X

Hands-on 3: Object properties

- Use the entity ontology you saved in the previous exercise
- Add an object property to model that people work for organisations
- Add a property value to model that you work for the organisation you specified previously
- Save the ontology

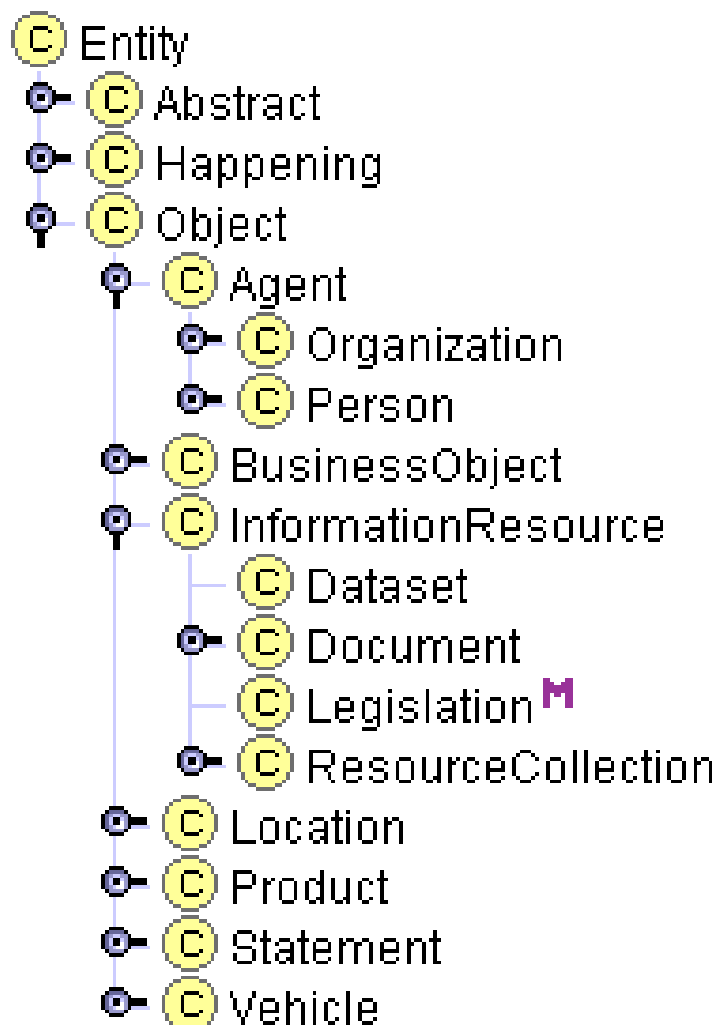
Ontology Design Principles

- There are many ways to encode a domain in an ontology – use your application needs as a guide
- Ontology authoring is often iterative and evolves with your text analysis application
- Classes vs instances: this can vary, but as a rough guide, proper nouns are usually instances, common nouns are usually classes
- Level of granularity: what subclasses do you need?
 - e.g do organisations need subclasses such as government, education, charity, etc?
- Domains and ranges:
 - Make sure they are generic enough, but not too generic
 - Information is propagated downwards, so don't add both a class and its subclasses as domain/range
 - Avoid using unions of classes for domains or ranges when a common superclass is available

PROTON Ontology

- a lightweight upper-level ontology developed by Ontotext
- 250 NE classes
- 100 relations and attributes
- covers mostly **NE classes**, and ignores general concepts

<http://proton.semanticweb.org>



Semantic Annotation

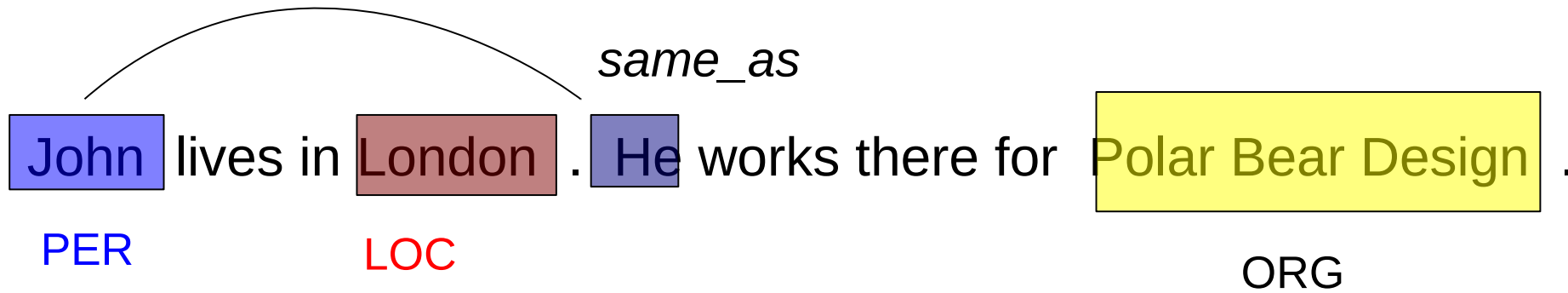
Information Extraction for the Semantic Web

- Traditional IE is based on a flat structure, e.g. recognising Person, Location, Organisation, Date, Time etc.
- For the Semantic Web, we need information in a hierarchical structure
- Idea is that we attach semantic metadata to the documents, pointing to concepts in an ontology
- Information can be exported as an ontology annotated with instances, or as text annotated with links to the ontology

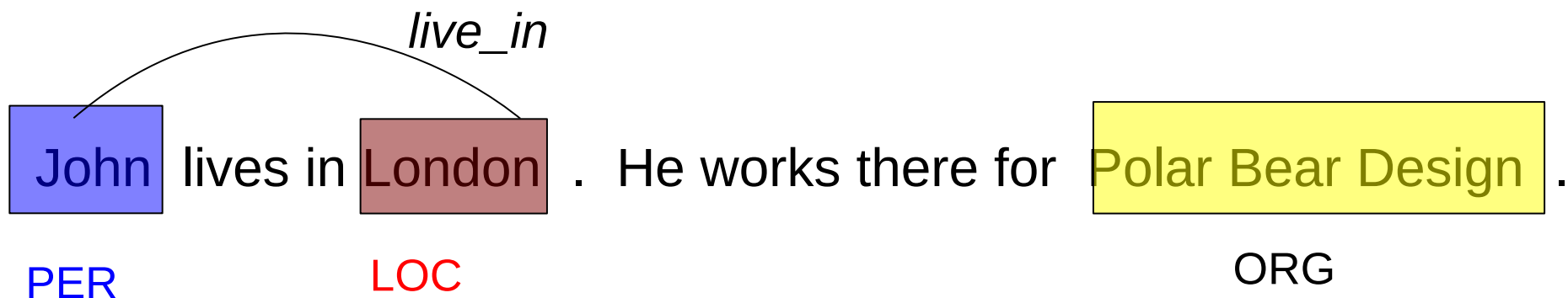
Traditional NE Recognition

John lives in London . He works there for Polar Bear Design .
PERSON LOCATION ORGANISATION

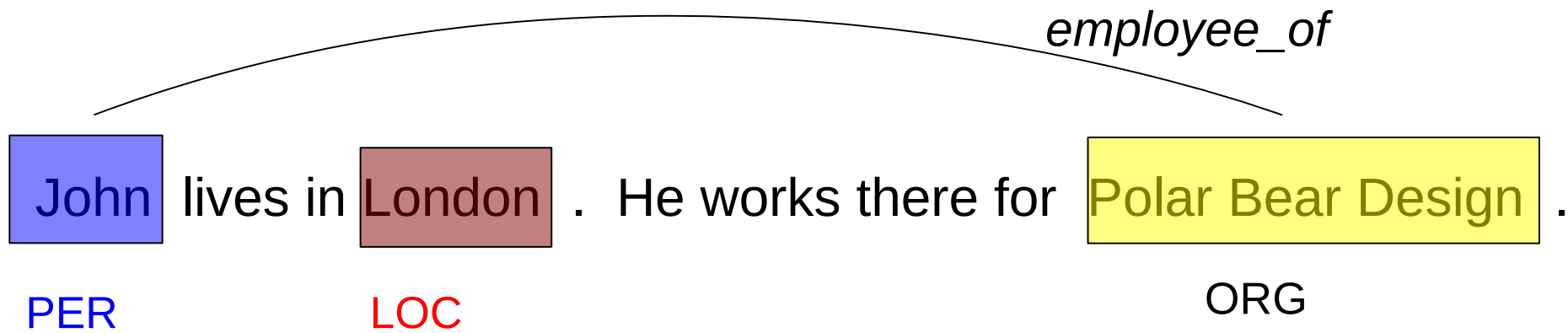
Co-reference



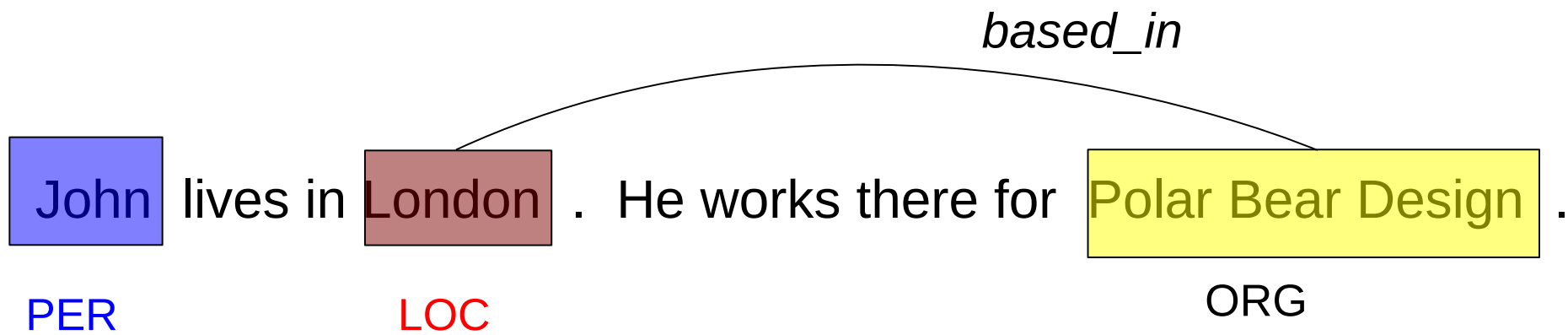
Relations



Relations (2)

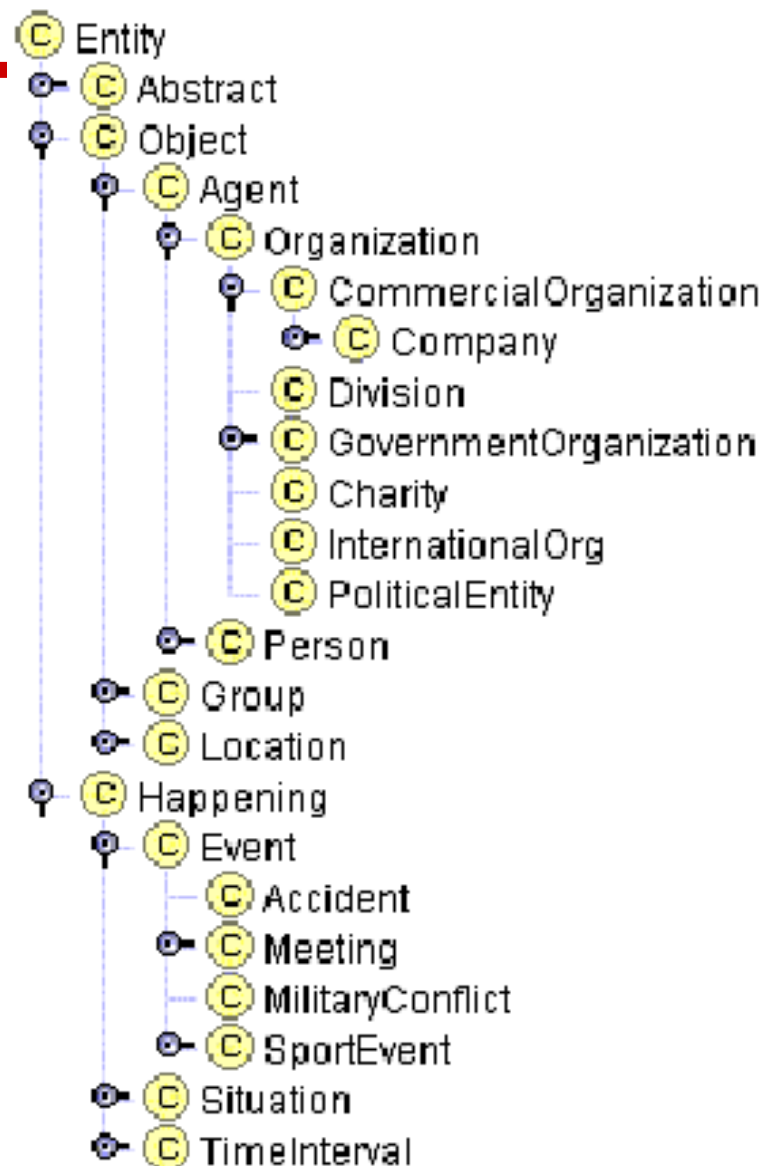


Relations (3)

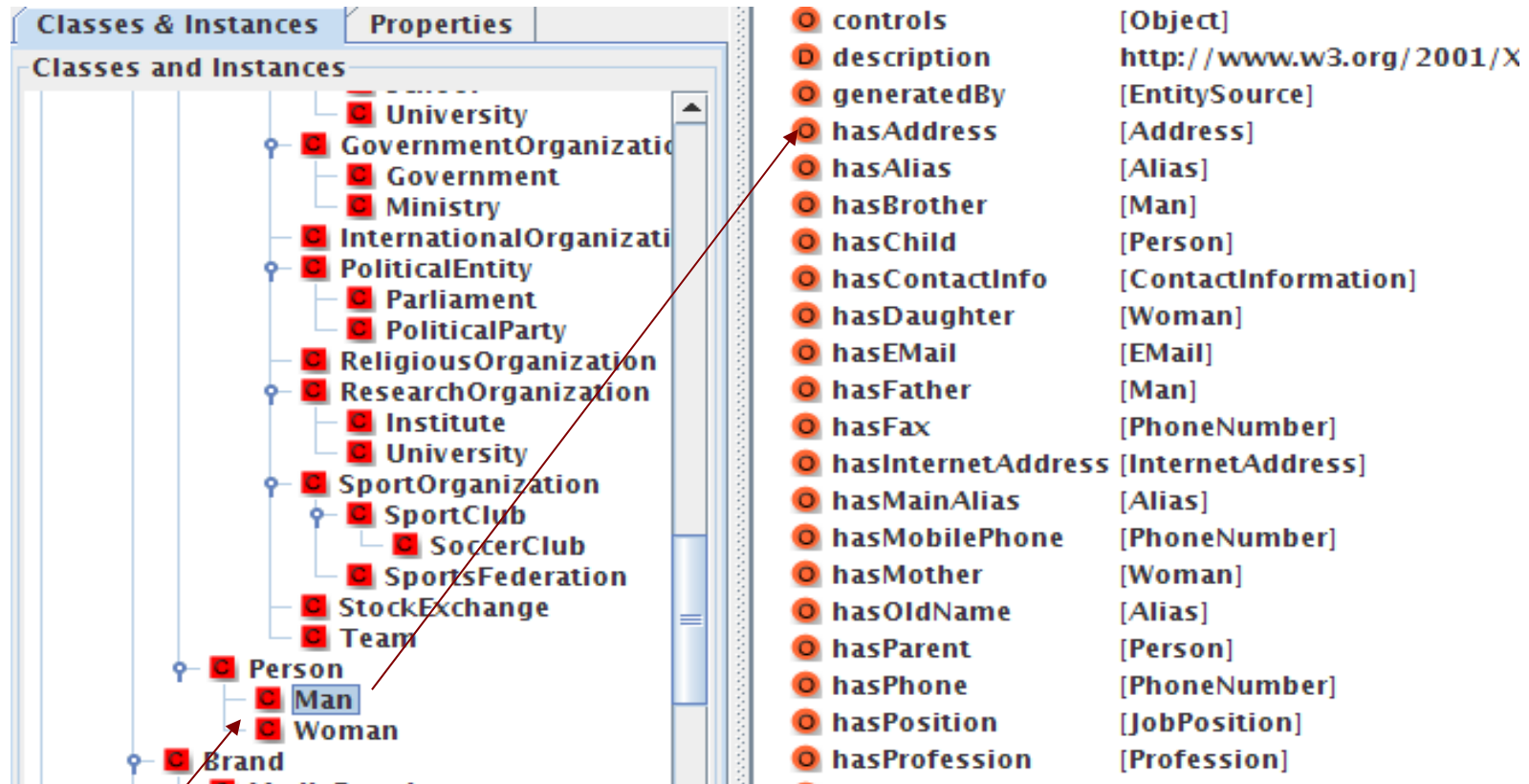


Richer NE Tagging

- Attachment of instances in the text to concepts in the domain ontology
- Disambiguation of instances, e.g. Cambridge, MA vs Cambridge, UK



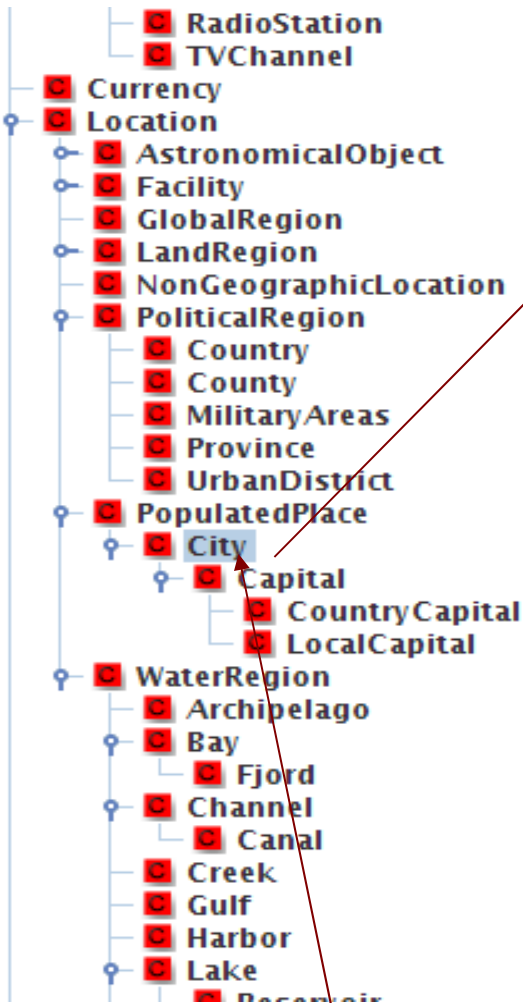
Ontology-based IE



John lives in London. He works there for Polar Bear Design.



Ontology-based IE (2)



hasAddress	[Address]
hasAirport	[Airport]
hasAlias	[Alias]
hasContactInfo	[ContactInformation]
hasEmail	[Email]
hasFax	[PhoneNumber]
hasInternetAddress	[InternetAddress]
hasMainAlias	[Alias]
hasMobilePhone	[PhoneNumber]
hasOldName	[Alias]
hasPhone	[PhoneNumber]
hasStationaryPhone	[PhoneNumber]
hasUniversity	[University]
hasWebPage	[WebPage]
isDefinedBy	[ALL RESOURCES]
isOwnedBy	[Agent]
label	[ALL RESOURCES]
laconicDescription	http://www.w3.org/2001/XMLSchema#string
latitude	http://www.w3.org/2001/XMLSchema#float
locatedIn	[Location]
longitude	http://www.w3.org/2001/XMLSchema#float
mainLabel	http://www.w3.org/2001/XMLSchema#string
partOf	[Entity]
populationCount	http://www.w3.org/2001/XMLSchema#integer
seeAlso	[ALL RESOURCES]
subRegionOf	[Location]

John lives in **London**. He works there for Polar Bear Design.

How does ontology-based IE help with IE?

- We can make inferences about all kinds of things once we have the annotations linked to an ontology
- We now know that cities can have airports, and people have phone numbers
- Since John is a man, we know he can have a wife
- If we know that the London, where John lives, is in England, we know that Polar Bear Design is also in England and not Ontario

Ontologies are useful for encoding the information found

- Enable us to define the concepts we're trying to find in texts
 - e.g., *aircraft accident, industrial action*
- As well as particular instances of these
 - e.g., *Qantas flight XYZ crashed on ..., BA cabin crew were on strike between March 20-23, 2010*
- And the relationships between them
 - e.g., *the plane that crashed belonged to Qantas and crashed on a specific date*

Using knowledge from the ontology

- The ontology tells us that
 - *Industrial action involves airport or airline staff and has a start and end date*
- It gives a clearly defined schema to annotate against
 - *if you annotate an instance of a strike, then you know this also requires you to annotate the airport/airline affected and the staff on strike*
- Extra knowledge about the different kinds of properties and the actors involved can help to improve system performance
- Backbone for other processes, for example visualising results on a timeline

Text mining and semantic annotation

- Extract **structured data** from text by
 - Linking references to entities
 - Linking entities to their semantic descriptions
- Automatic **semantic annotation** based on IE technology
- Attaches **metadata** to documents, which makes them more useful and more easily processable
- They can then be used for searching and hyperlinking, categorising, and monitoring
- Adds value to content of libraries, enabling user **interaction** with content
- Enhanced capability for cross-referencing and **dynamic** document classification

Some Terminology

- **Semantic annotation** – annotate in the texts all mentions of instances relating to concepts in the ontology
- **Ontology learning** – automatically derive an ontology from texts
- **Ontology population** – given an ontology, populate the concepts with instances derived automatically from a text

Semantic Annotation vs Ontology Population

- **Semantic Annotation**
 - Mentions of instances in the text are annotated wrt concepts (classes) in the ontology.
 - Requires that instances are disambiguated.
 - It is the **document** which is modified.
- **Ontology Population**
 - Generates new instances in an ontology from a text.
 - Links unique mentions of instances in the text to instances of concepts in the ontology.
 - Instances must be not only disambiguated but also co-reference between them must be established.
 - It is the **ontology** which is modified.



Semantic Annotation

Print

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, t
and final group match.

But the day of decision for the G
stave off default by honouring bo

The EU and the IMF have agreed

Location

class	http://dbpedia.org/ontology/Place	X
inst	http://dbpedia.org/resource/Brussels	X
locType	other	X
matches	[6413, 6412]	X
rule	LKB_Location	X
		X

► Open Search & Annotate tool

☒ Content

☐ Date

☐ Document

☐ DocumentClassification

☐ DocumentDate

☐ DocumentTitle

☐ FirstPerson

☐ JobTitle

☒ Location

☐ Lookup

☐ Measurement

☐ Money

☐ Number

☒ Organization

☐ Person

☐ Ratio

☐ Sentence

☐ SpaceToken

☐ Split

☐ Temp

☐ Title

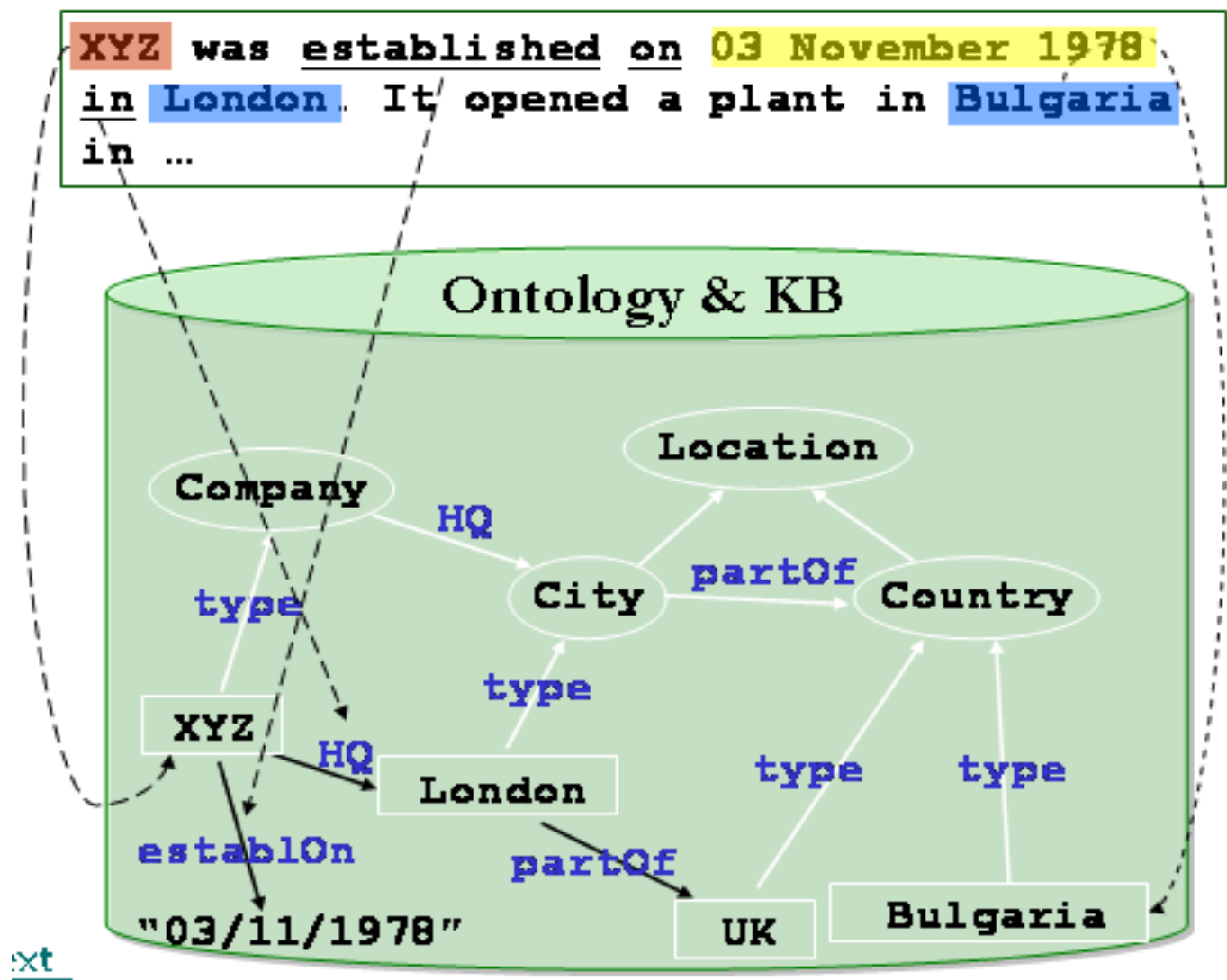
☐ Token

☐ Unknown

► Original markups

Type	Set	Start	End
Location		1222	1228
Location		1222	1228
Location		1222	1228
Location		1222	1228
Location		1222	1228
Location		1222	1228
Location		1222	1228
Location		1233	1241
Organization		1556	1558

Ontology Population



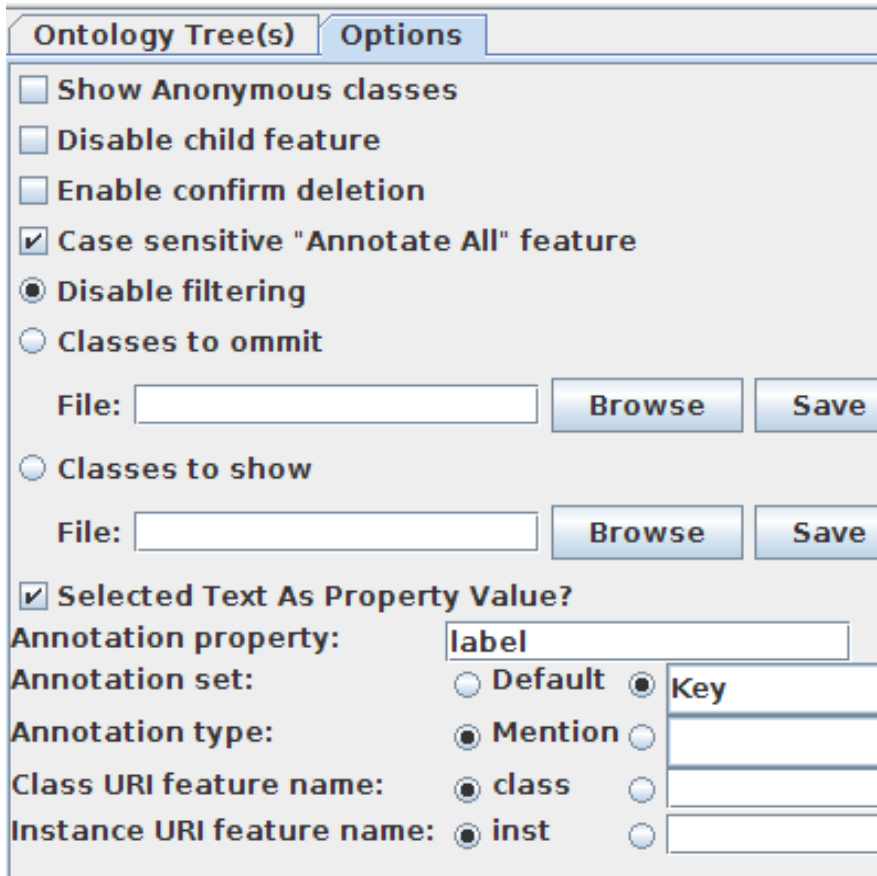
How to do Semantic Annotation

- Manually: ontology based annotation using OAT (Ontology Annotation Tool)
- Automatically
 - Gazetteer/rule/pattern based
 - Classifier (ML) based
 - Combination of the two

Manual semantic annotation: OAT

- Shows document and ontology class hierarchy side-by-side
- Interactive creation of annotations that link to the ontology class/instance
- Allows on-the-fly instance creation
- Used to create evaluation or training corpus

OAT: Options Tab



The screenshot shows the 'Options' tab of the OAT (Ontology Annotation Tool) interface. It contains several configuration options for document annotation.

Options Tab

- ☐ Show Anonymous classes
- ☐ Disable child feature
- ☐ Enable confirm deletion
- ☒ Case sensitive "Annotate All" feature
- ☒ Disable filtering
- ☐ Classes to omit
- File:
- ☐ Classes to show
- File:
- ☒ Selected Text As Property Value?

Annotation property:

Annotation set: ☐ Default ☒ Key

Annotation type: ☒ Mention ☐

Class URI feature name: ☒ class ☐

Instance URI feature name: ☒ inst ☐

- Customisation has to be done for each document
- To ensure that any new instances automatically have a label (the string you selected in the document), tick *Select text as property value*.
- To put all annotations into a set other than Default, change accordingly
- By default, OAT creates:
 - Annotations of type *Mention*
 - *class* feature with the class URI
 - *inst* feature with the instance URI



OAT

As well as picking MPs for Westminster, voters will elect councillors in 164 local authorities across England.

Voting in the general election will take place in 649 constituencies, with nearly 4,150 candidates standing for election across the country.

David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

Labour leader Gordon Brown went to vote shortly after 1100 BST at a community centre close to his home in North Queensferry, Fife. His wife Sarah was with him.

Nick Clegg, leader of the Liberal Democrats, arrived at a polling station in Sheffield Hallam at 1120 BST. His wife Miriam is unable to vote in the general election because she is a Spanish citizen.

The leader of the Scottish National Party, Alex Salmond, cast his vote shortly before noon, at Macduff in Banffshire. Ieuan Wyn Jones, leader of the Welsh Labour Party, was seen in the constituency of Ynys Mon in north Wales.

Polling in one constituency - Thirsk and Market Rasen - was delayed in May because of the death of one of the candidates. [ELECTION 2010 ON THE BBC](#)

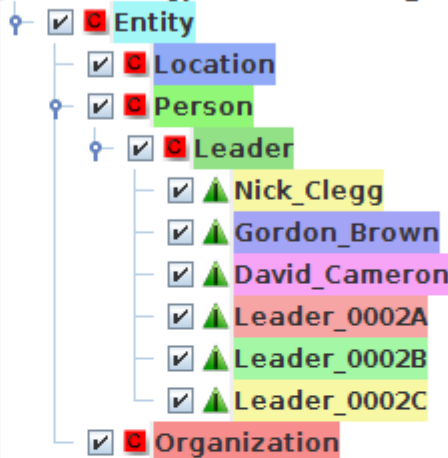
Type	Set	Start	End	Id	
Mention		1277	1289	55	{ class=http

Ontology Tree(s)

Options

test-ontology-instances.owl_00018

test-ontology-instances.owl_00018

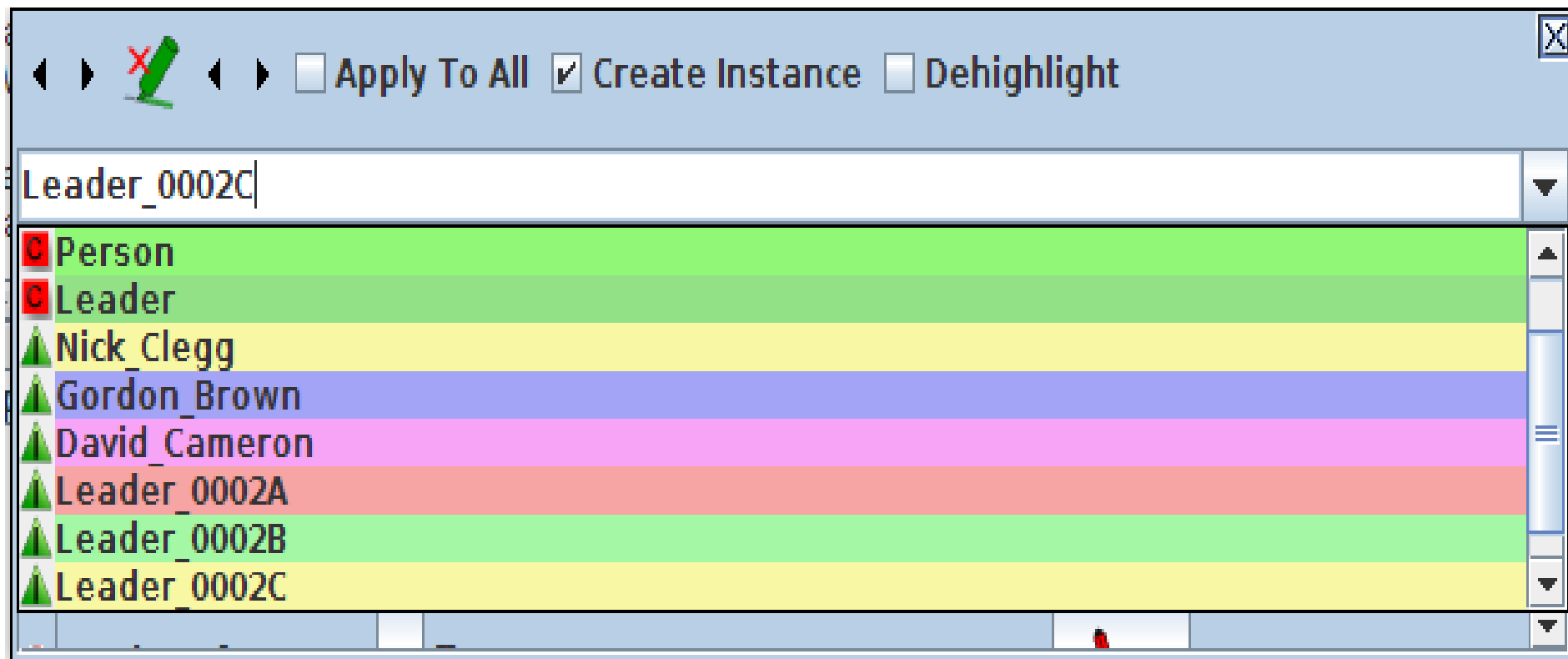


Apply To All Create Instance Dehighlight

Leader_0002C

- Person
- Leader
- Nick_Clegg
- Gordon_Brown
- David_Cameron
- Leader_0002A
- Leader_0002B
- Leader_0002C

OAT: The Editor Pop-up

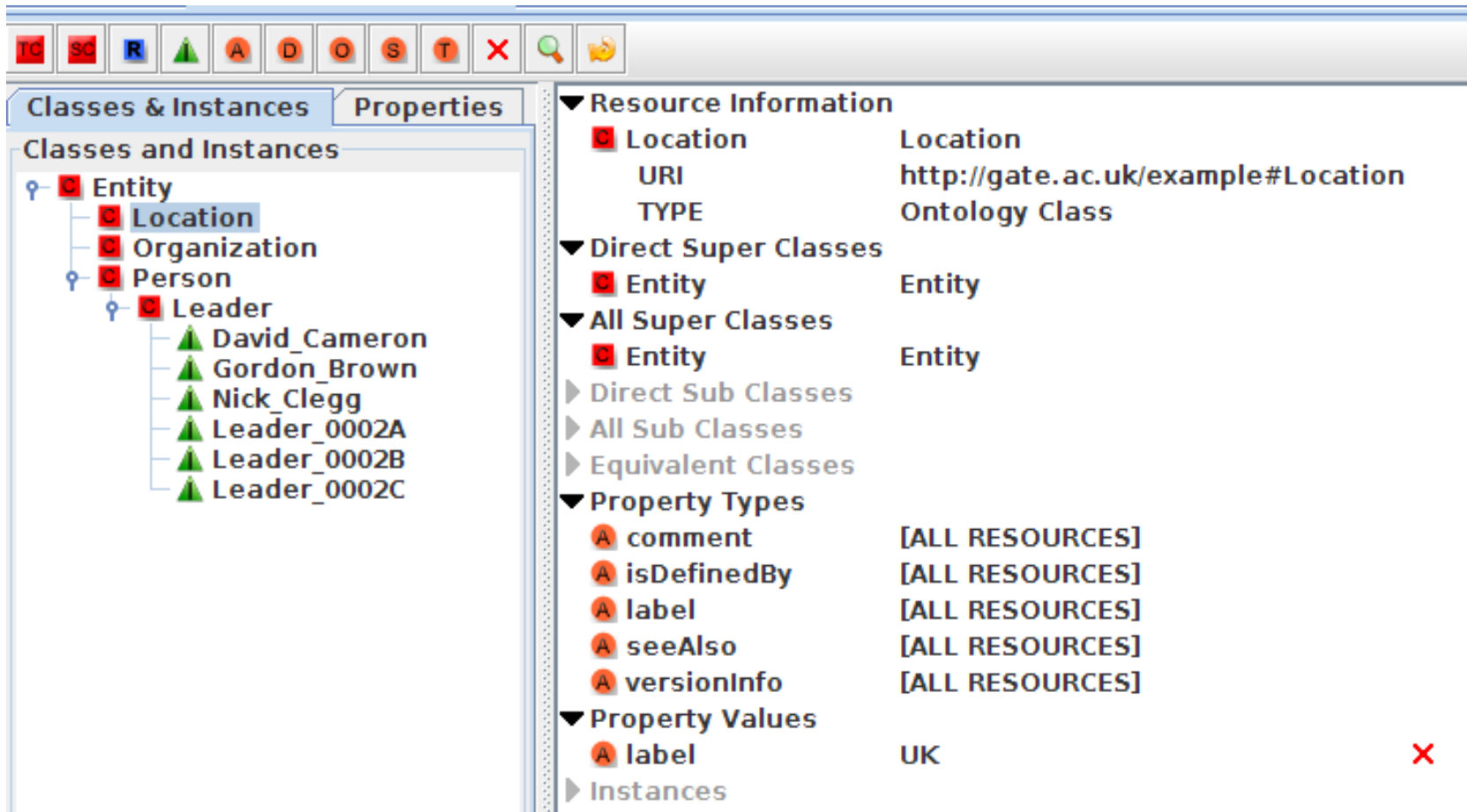


Annotating classes and instances

- Be very careful about the difference between annotating classes and instances
- If you want to add **UK** to the ontology as an instance of a **Location**, you need to select “Create instance”
- Note that this will create a new instance in the ontology with a name like **Location_00020**. The string **UK** will appear as a label on that instance.
- If you just want to annotate **UK** with the class **Location**, then deselect “Create instance”

Annotating a class

Annotating **UK** as a **class** will create a new label on the class with the text string



The screenshot shows the GATE interface with the 'Classes & Instances' panel on the left and the 'Properties' panel on the right.

Classes and Instances

- Entity
 - Location
 - Organization
 - Person
 - Leader
 - David_Cameron
 - Gordon_Brown
 - Nick_Clegg
 - Leader_0002A
 - Leader_0002B
 - Leader_0002C

▼ Resource Information

Location	Location
URI	http://gate.ac.uk/example#Location
TYPE	Ontology Class

▼ Direct Super Classes

Entity	Entity
--------	--------

▼ All Super Classes

Entity	Entity
--------	--------

► Direct Sub Classes

► All Sub Classes

► Equivalent Classes

▼ Property Types

comment	[ALL RESOURCES]
isDefinedBy	[ALL RESOURCES]
label	[ALL RESOURCES]
seeAlso	[ALL RESOURCES]
versionInfo	[ALL RESOURCES]

▼ Property Values

label	UK
-------	----

► Instances





Annotating an instance

Classes & Instances

Properties

Classes and Instances

Entity

Location

Location_0002D

Organization

Person

Leader

David_Cameron

Gordon_Brown

Nick_Clegg

Leader_0002A

Leader_0002B

Leader_0002C

▼ Resource Information

▲ Location_0002D

Location_0002D

URI

http://gate.ac.uk/example#Location_0002D

TYPE

Ontology Instance

▼ Direct Types

■ Location

Location

▼ All Types

■ Location

Location

■ Entity

Entity

▶ Same Instances

▼ Property Types

▲ seeAlso

[ALL RESOURCES]

▲ versionInfo

[ALL RESOURCES]

▲ comment

[ALL RESOURCES]

▲ label

[ALL RESOURCES]

▲ isDefinedBy

[ALL RESOURCES]

▼ Property Values

▲ label

UK

X

Hands-on 4: using OAT

- Use the previously created ontology
- Load the document **voting-example.xml** (from hands-on)
- Select the OAT button from the doc viewer
- From the Options tab, choose Key as the annotation set and tick “Select text as property value”
- Annotate every instance of UK in the text as an instance of a Location
 - **Tip:** Make sure you select “Create instance” and “Apply to all” before choosing the target class
- Switch to the ontology viewer to see the new instance
- Examine the annotations created in Key and their features
- Save the ontology and the document

OAT: comments

- The options to filter out some classes or only show some are useful when working with big ontologies
- Limitations: cannot annotate property values
- Solution: RAT (Relation Annotation Tool)

Relation Annotation Tool (RAT)

- RAT annotates a document with ontology instances and creates relations between annotations by means of ontology object properties.
- It is compatible with OAT, but focuses on relations between annotations
- You need the Ontology Tools plugin loaded
- It is comprised of 2 viewers: **RATC** (RAT-Concept) and **RATI** (Rat-Instance). You need both open simultaneously.
- In the document editor, click on the **RATC** and **RATI** buttons to enable the viewers
- The RATC pane (on the RHS) looks similar to OAT. Click the checkbox beside a class to display the relevant instances.

RAT-I: Adding Instances and Properties

- The RAT-I view (lower horizontal pane) shows two columns: one for instances and one for properties
- To create a new instance, select an item in the ontology and then select the relevant text in the document
- Click “New instance”
- Any properties on the relevant class will be shown on the RHS of the table
- To add a property range, select a property and choose a value from the dropdown list
- Only object properties will be shown: it is not possible to add datatype properties in this way

Hands-on 5: RAT

- Use the document from the previous hands-on
- Load the ontology test-ontology-instances.owl and remove the old ontology
- Click on RAT-C and RAT-I to display the viewers
- Add a new instance **Liberal Democrats** to the class **Organization**
- Add a new instance **Nick Clegg** to the class **Leader**
- Select the **Nick Clegg** instance and add the value of the **works_for** property to **Liberal Democrats**
- Use the ontology viewer to check the results, then save the ontology.
- Tip: select the relevant concepts using the checkboxes in the RAT-C pane to display the instances in the RAT-I pane



Adding a property value

Your result should look something like this:

close to his home in North Queensberry, Fife. His wife Sarah was with him.

Nick Clegg, leader of the Liberal Democrats, arrived at a polling station in Sheffield Hallam at 1120 BST. His wife Miriam is unable to vote in the general election because she is a Spanish citizen.

The leader of the Scottish National Party, Alex Salmond, cast his vote shortly before noon, at Macduff in Banffshire. Ieuan Wyn Jones of Plaid Cymru voted in the constituency of Ynys Mon in north Wales at lunchtime.

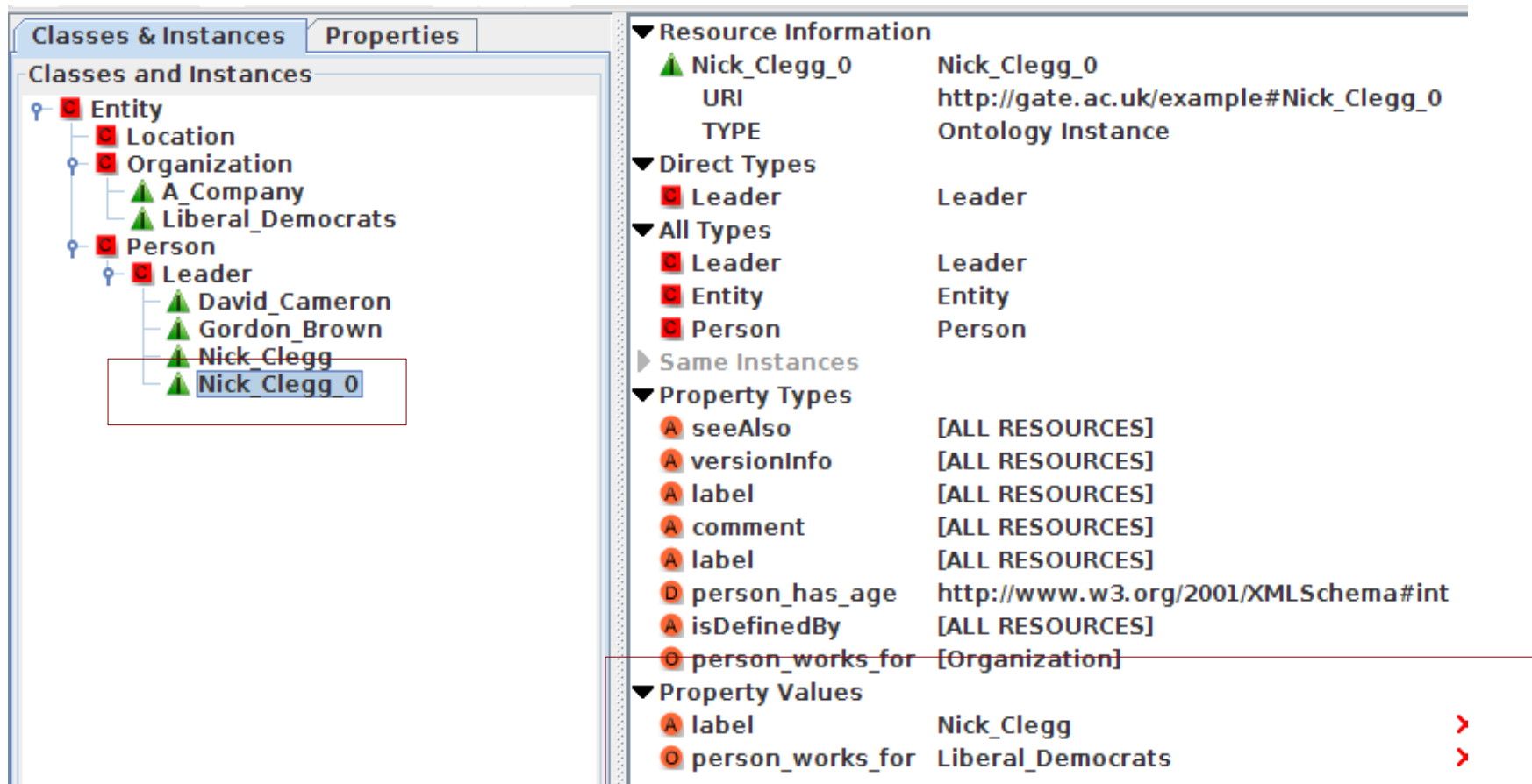
Filter: X

New Inst. Add to Selected Inst.

Instance	Label	Property	Value
Nick_Clegg_0	[Nick_Clegg]	person_works_for	[Organization]
		person_works_for	Liberal_Democrats

Checking the result

Check that the instance and property have been added correctly, by viewing it in the ontology editor



The screenshot displays the GATE ontology editor interface, showing the 'Classes & Instances' and 'Properties' tabs.

Classes & Instances:

- Entity
 - Location
 - Organization
 - A_Company
 - Liberal_Democrats
 - Person
 - Leader
 - David_Cameron
 - Gordon_Brown
 - Nick_Clegg
 - Nick_Clegg_0

Properties:

- Resource Information
 - Nick_Clegg_0
 - URI: http://gate.ac.uk/example#Nick_Clegg_0
 - TYPE: Ontology Instance
- Direct Types
 - Leader
- All Types
 - Leader
 - Entity
 - Person
- Same Instances
- Property Types
 - seeAlso: [ALL RESOURCES]
 - versionInfo: [ALL RESOURCES]
 - label: [ALL RESOURCES]
 - comment: [ALL RESOURCES]
 - label: [ALL RESOURCES]
 - person_has_age: <http://www.w3.org/2001/XMLSchema#int>
 - isDefinedBy: [ALL RESOURCES]
 - person_works_for: [Organization]
- Property Values
 - label: Nick_Clegg
 - person_works_for: Liberal_Democrats

OAT vs RAT

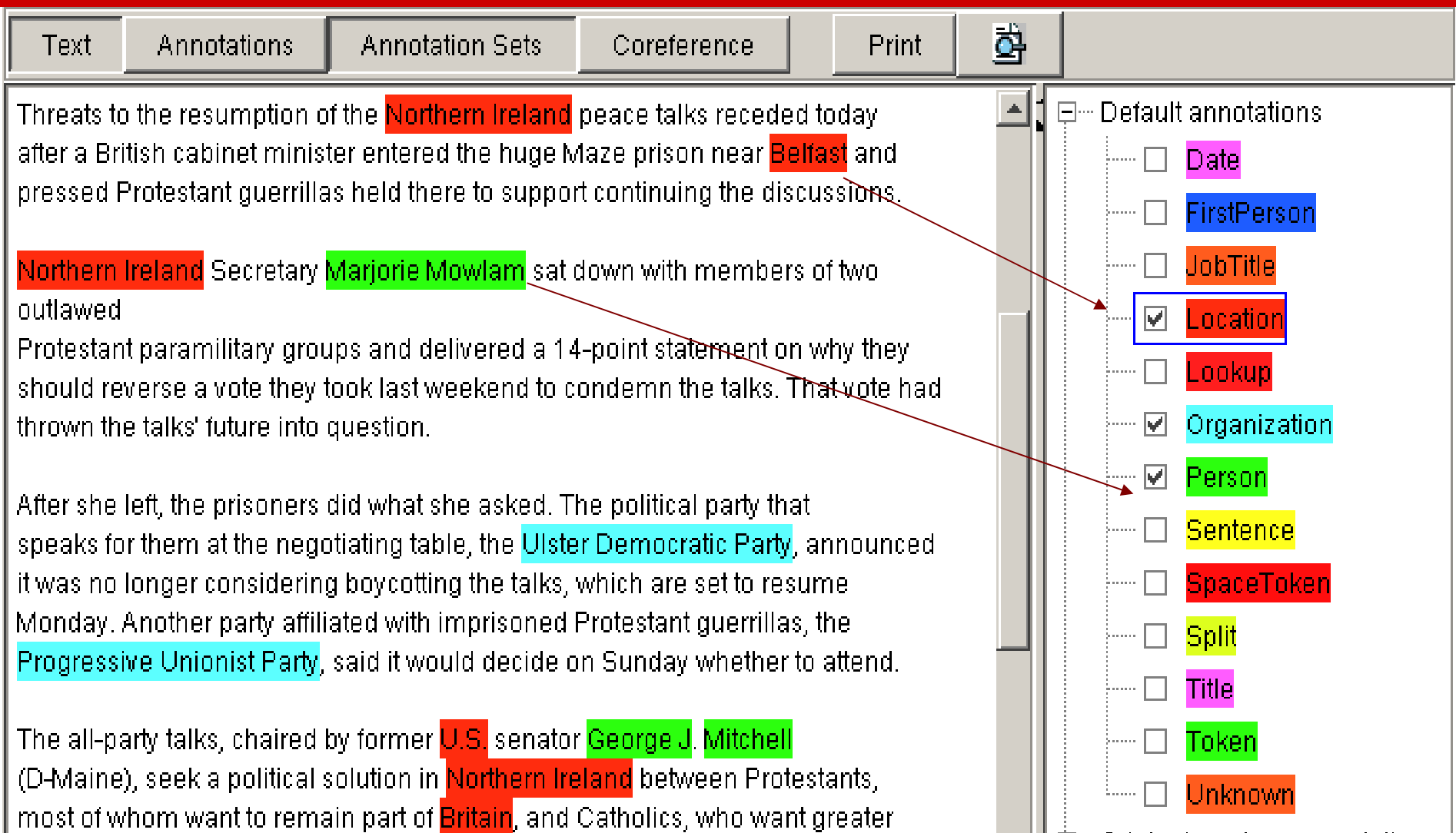
- In OAT, you have the option to annotate all mentions of the selected string in one go, e.g. the string “Liberal Democrats” as being the mention of the respective instance from the ontology. In RAT, you'll have to annotate each of the occurrences of this string over and over again
- OAT currently creates rather opaque instance URIs (e.g., Leader_0007A with label “David Cameron”), so once you have several automatically created instances of the same class, it becomes hard to distinguish which is which in OAT. RAT shows you all labels, not just the URI, so it's easier to select
- In OAT you can annotate a string as a mention of a class, without giving an instance



Automatic Semantic Annotation in GATE

- GATE supports ontologies as part of IE applications - Ontology-Based IE (OBIE)
- Supports semantic annotation and ontology population
- GATE has its own ontology API based on Sesame 2 and OWLIM 3
- Semantic annotation can combine learning and rule-based methods
- Enables use of large-scale linguistic resources for IE, such as WordNet

Traditional IE in GATE



The screenshot displays the GATE software interface. At the top, there are tabs for 'Text', 'Annotations', 'Annotation Sets', 'Coreference', and 'Print'. The 'Text' tab is active, showing a document with several paragraphs. The text is annotated with colored boxes representing named entities. On the right side, there is a panel titled 'Default annotations' with a list of annotation types, each with a checkbox. The 'Location' checkbox is checked and highlighted with a blue box. The 'Organization' and 'Person' checkboxes are also checked. Red arrows point from the 'Location' checkbox to the 'Northern Ireland' and 'Belfast' annotations in the text. Another red arrow points from the 'Person' checkbox to the 'Marjorie Mowlam' annotation. The text in the document is as follows:

Threats to the resumption of the Northern Ireland peace talks receded today after a British cabinet minister entered the huge Maze prison near Belfast and pressed Protestant guerrillas held there to support continuing the discussions.

Northern Ireland Secretary Marjorie Mowlam sat down with members of two outlawed Protestant paramilitary groups and delivered a 14-point statement on why they should reverse a vote they took last weekend to condemn the talks. That vote had thrown the talks' future into question.

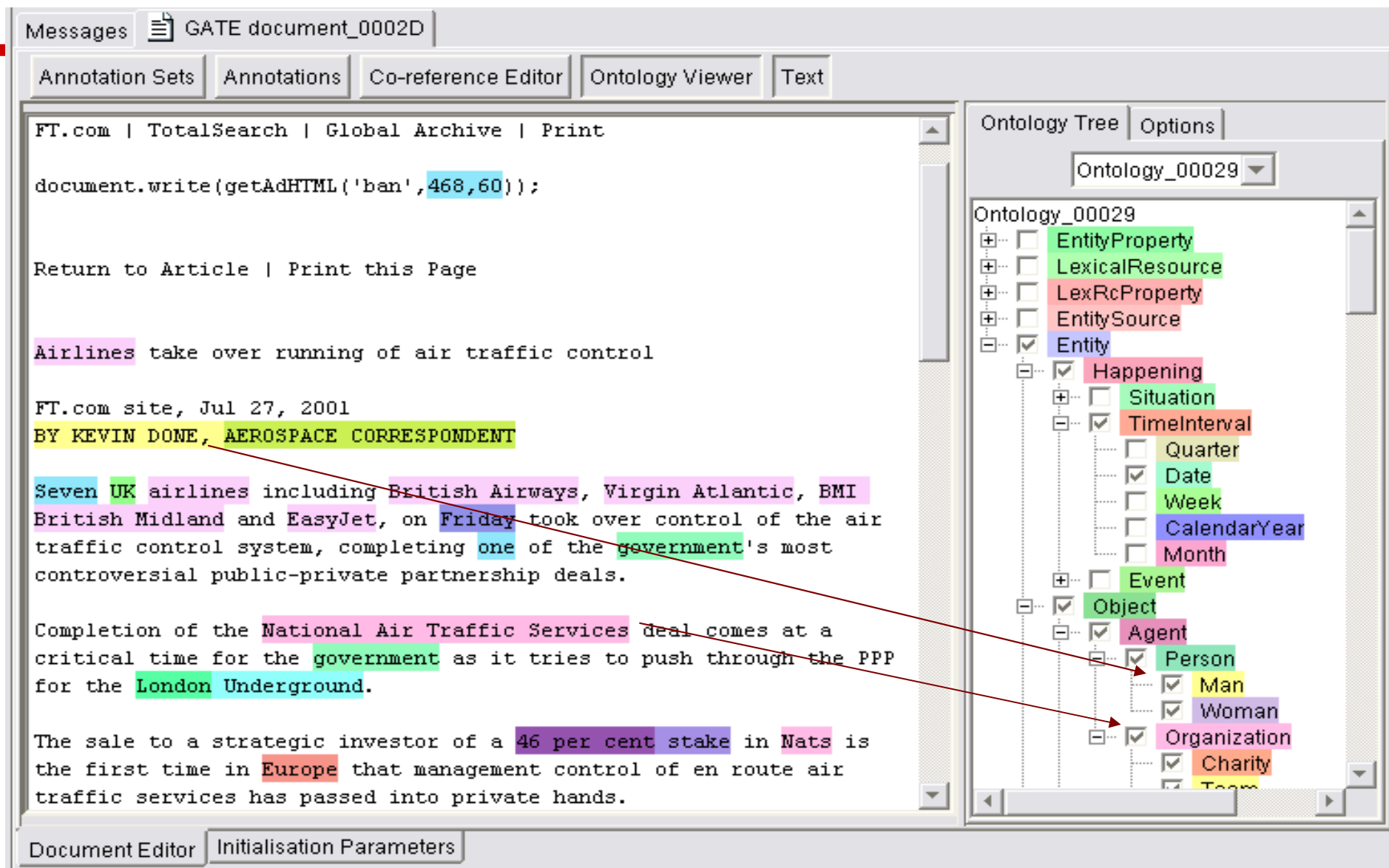
After she left, the prisoners did what she asked. The political party that speaks for them at the negotiating table, the Ulster Democratic Party, announced it was no longer considering boycotting the talks, which are set to resume Monday. Another party affiliated with imprisoned Protestant guerrillas, the Progressive Unionist Party, said it would decide on Sunday whether to attend.


The all-party talks, chaired by former U.S. senator George J. Mitchell (D-Maine), seek a political solution in Northern Ireland between Protestants, most of whom want to remain part of Britain, and Catholics, who want greater

Default annotations

- ☐ Date
- ☐ FirstPerson
- ☐ JobTitle
- ☒ Location
- ☐ Lookup
- ☒ Organization
- ☒ Person
- ☐ Sentence
- ☐ SpaceToken
- ☐ Split
- ☐ Title
- ☐ Token
- ☐ Unknown

Semantic IE in GATE



Messages  GATE document_0002D

Annotation Sets Annotations Co-reference Editor Ontology Viewer Text

FT.com | TotalSearch | Global Archive | Print

```
document.write(getAdHTML('ban',468,60));
```

Return to Article | Print this Page

Airlines take over running of air traffic control

FT.com site, Jul 27, 2001

BY KEVIN DONE, AEROSPACE CORRESPONDENT

Seven UK airlines including British Airways, Virgin Atlantic, BMI British Midland and EasyJet, on Friday took over control of the air traffic control system, completing one of the government's most controversial public-private partnership deals.

Completion of the National Air Traffic Services deal comes at a critical time for the government as it tries to push through the PPP for the London Underground.

The sale to a strategic investor of a 46 per cent stake in Nats is the first time in Europe that management control of en route air traffic services has passed into private hands.

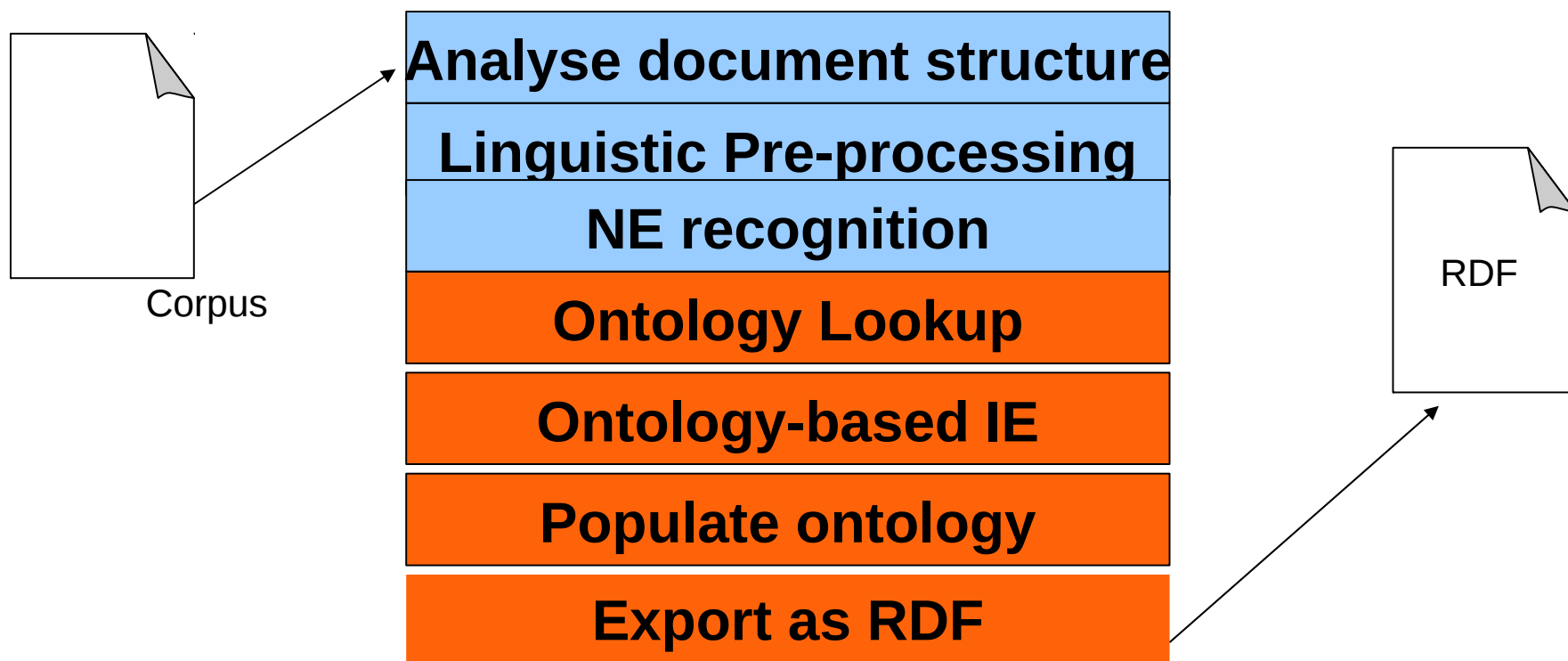
Ontology Tree Options

Ontology_00029

- EntityProperty
- LexicalResource
- LexRcProperty
- EntitySource
- ☒ Entity
 - ☒ Happening
 - ☒ Situation
 - ☒ TimeInterval
 - Quarter
 - ☒ Date
 - Week
 - CalendarYear
 - Month
 - ☒ Event
 - ☒ Object
 - ☒ Agent
 - ☒ Person
 - ☒ Man
 - ☒ Woman
 - ☒ Organization
 - ☒ Charity
 - ☒ Team

Document Editor Initialisation Parameters

Typical Semantic Annotation pipeline



Ontology Lookup: OntoRoot Gazetteer

- Finds mentions in the text matching classes, instances, data property values and labels in the ontology
- Matching can be done between any morphological or typographical variant (e.g. upper/lower case, CamelCase)
- Converts CamelCase names, hyphens, underscores
- Morphological analysis is performed on both text and ontology, then matching is done between the two at the root level.
- Text is annotated with features containing the root and original string(s)
- Creates a gazetteer PR that can be used with the FlexibleGazetteerPR

OntoRoot Gazetteer

- Lives in the Gazetteer_Ontology_Based plugin
- Generates candidate gazetteer list from ontology
- Runs the Tokeniser, POS tagger, Morphological Analyser to create lemmas from the labels and the URIs of all classes and instances and then creates lists to match against the text
 - Gordon_Brown → Gordon Brown
- Note that the gazetteer produced is stored in memory only and cannot be edited
- The OntoRoot gazetteer must always be preceded by tokeniser, sentence splitter (required by the POS tagger), POS tagger and a morphological analyser in the pipeline



Init-time OntoRoot params

Parameters for the new Onto Root Gazetteer

Name:

Name	Type	Required	Value
caseSensitive	java.lang.Boolean	<input checked="" type="checkbox"/>	<input type="text" value="false"/>
considerHeuristicRules	java.lang.Boolean	<input checked="" type="checkbox"/>	<input type="text" value="false"/>
considerProperties	java.lang.Boolean	<input checked="" type="checkbox"/>	<input type="text" value="true"/>
morpher	gate.creole.morph.Morph	<input checked="" type="checkbox"/>	MorphAnal
ontology	gate.creole.ontology.Ontology	<input checked="" type="checkbox"/>	<input type="text" value="<none>"/>
posTagger	gate.creole.POSTagger	<input checked="" type="checkbox"/>	<input type="text" value="<none>"/>
propertiesToExclude	java.lang.String	<input type="checkbox"/>	<input type="text"/>
propertiesToInclude	java.lang.String	<input type="checkbox"/>	<input type="text"/>
separateCamelCasedWords	java.lang.Boolean	<input checked="" type="checkbox"/>	<input type="text" value="true"/>
tokeniser	gate.creole.tokeniser.DefaultTokeniser	<input checked="" type="checkbox"/>	<input type="text" value="<none>"/>
useResourceUri	java.lang.Boolean	<input checked="" type="checkbox"/>	<input type="text" value="true"/>

OK

Help

Cancel

Ontology LR

POS Tagger

Tokeniser

Running the OntoRoot gazetter

- If mostly matching proper names, then add to your application and run like the ANNIE gazetteer
- It will match against the document text as it is, which is not ideal if matching against terms
- We recommend building an application which also tokenises, POS tags, and morphologically analyses the text
- Then load the Flexible Gazetteer PR and provide OntoRoot as the gazetteer to run
- Note that it is the Flexible Gazetteer and not the OntoRoot Gazetteer that you need to add to the application
- The Flexible Gazetteer just calls the OntoRoot Gazetteer



OntoRoot Application in GATE

Create a Flexible Gazetteer with an OntoRoot inside it

Parameters for the new Flexible Gazetteer

Name:

Name	Type	Required	Val
gazetteerInst	Gazetteer	✓	Onto Root Gazetteer_02277
inputFeatureNames	List	✓	[Token.root]

OK Cancel Help

Build a GATE application with the PRs shown

Selected Processing resources

!	Name	Typ
	ANNIE English Tokeniser_00077	ANNIE English
	ANNIE Sentence Splitter_0228C	ANNIE Sentenc
	ANNIE POS Tagger_0007B	ANNIE POS Tag
	GATE Morphological analyser_0007A	GATE Morpholo
	Flexible Gazetteer_02291	Flexible Gazet

Output Example

standing for election across the country.

David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

Lookup

URI	http://gate.ac.uk/example#	
classURI	http://gate.ac.uk/example#	
classURIList	[http://gate.ac.uk/example#	
heuristic_level	0	
majorType		
type	instance	

Open Search & Annotate tool

Lookup

URI	http://gate.ac.uk/example#Leader	X
heuristic_level	0	X
majorType		X
type	class	X
		X

Open Search & Annotate tool

Lookup	672	685	9704	{URI=http://gate.ac.uk/example#David_Cameron, classURI=http://g
Lookup	721	728	9705	{URI=http://gate.ac.uk/example#Leader, heuristic_level=0, majorTy
Lookup	758	764	9706	{URI=http://gate.ac.uk/example#Leader, heuristic_level=0, majorTy

- The URI feature contains the matched class or instance URI
- The type feature is either class or instance
- Instances have also a feature classURI

Hands-on 6: OntoRootGazetteer

- Load Gazetteer_Ontology_Based plugin, ANNIE and Tools plugin
- Close any open ontologies, but keep the document you have open
- Load the ontology **test-ontology-instances.owl**
- Create a new corpus pipeline
- Create Document Reset, Tokeniser, Sentence Splitter, POS Tagger, and Morphological Analyser (all with defaults) and add to the pipeline in that order
- Create separate Tokeniser, POS Tagger, and Morphological Analyser PRs which you use only as parameters for OntoRoot Gaz
- Create and configure OntoRootGazetteer with defaults and the ontology loaded

Hands-on 6: OntoRoot (contd.)

- Create Flexible Gazetteer
 - add OntoRootGazetteer as gazetteerInst
 - Specify Token.root for inputFeatureNames
- Add Flexible Gazetteer to the pipeline
- Set the runtime parameters of the Document Reset:
 - *setsToRemove* should contain *Test*
- Set all the input and output sets in the pipeline to *Test*
- Create a document from *voting-example.xml*
- Add it to a corpus and run the pipeline on this
- Inspect the resulting Lookup annotations in the ***Test*** annotation set
- Save your application for later use and keep it open

What next?

- Annotations from the ontology contain just candidate URIs and classes.
- These annotations might overlap
- We may need additional knowledge for
 - Disambiguation (e.g. where 2 people have the same name)
 - Semantic enrichment for subsequent processing stages (e.g. world knowledge encoded in other ontologies)

Conventions in GATE

- We use Mention annotations to reflect the fact that the text mentions a particular instance or a class
- The Mention annotations have two special features:
 - *class* = class URI from the ontology
 - *inst* = instance URI from the ontology (if available)
e.g. Mention {class=Leader, inst=Gordon_Brown}
- It's important not to use *class* and *inst* as features unless you're dealing with ontologies, as these are reserved names.

Ontology Aware JAPE

- JAPE transducers have a run-time parameter which is an ontology
- Note that NE transducers don't have this parameter, so you can't use them 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
- 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

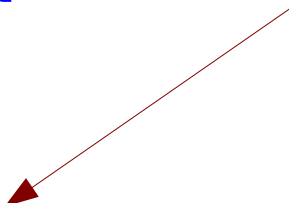
Input: Lookup

Options: control = appelt

Rule: PersonLookup

(
 {Lookup.class == Person}

Matches the class Person
or any of its subclasses

A red arrow originates from the text "Matches the class Person or any of its subclasses" and points diagonally down and to the left, ending at the condition "{Lookup.class == Person}" within the JAPE rule.

):person

-->

:person.Mention =
 {class = :person.Lookup.class,
 inst = :person.Lookup.inst}

Adds class and instance information
as features on the Mention annotation

A red arrow originates from the text "Adds class and instance information as features on the Mention annotation" and points diagonally down and to the left, ending at the feature assignment ":person.Mention = {class = :person.Lookup.class, inst = :person.Lookup.inst}".

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]}
```

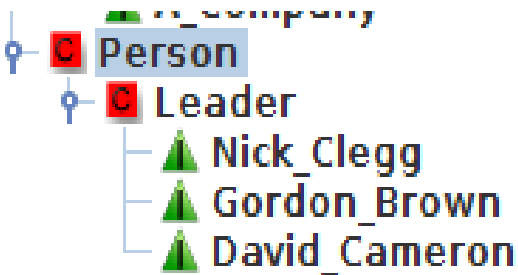
If you switch to a newer version of PROTON, you only need to change the Template declarations, not every JAPE LHS. (See the GATE User Guide <http://gate.ac.uk/userguide/sec:jape:templates> for more details and examples.)

```
Template: protont =  
    "http://proton.semanticweb.org/2006/05/protont#${n}"  
...
```



Matching subclasses

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



Lookup			
C URI	▼	http://gate.ac.uk/example#David_Cameron	▼ X
C class	▼	http://gate.ac.uk/example#Leader	▼ X
C classURI	▼	http://gate.ac.uk/example#Leader	▼ X
C classURIList	▼	[http://gate.ac.uk/example#Leader]	▼ X
C heuristic_level	▼	0	▼ X
C inst	▼	http://gate.ac.uk/example#David_Cameron	▼ X
C majorType	▼		▼ X
C type	▼	instance	▼ X

The rule matches because Leader is a subclass of Person

Compatibility with OntoRootGazetteer

- The OntoRootGazetteer puts the class URIs in a feature called **classURI** and the instance URI in a feature called **URI**
- But JAPE requires these features to be called **class** and **inst**
- So we need a JAPE grammar to first change the names of these features

JAPE grammar to change feature names

Phase: LookupRename

Input: Lookup

Options: control = appelt

Rule: RenameLookup

```
(  
  {Lookup.type == instance}
```

finds all Lookups with an **instance** feature

```
):match
```

```
-->
```

```
:match{
```

```
  for (Annotation lookup : matchAnnots) {  
    FeatureMap theFeatures = lookup.getFeatures();
```

add a new feature **class** with the
value of the original classURI feature

```
    theFeatures.put(  
      "class", theFeatures.get("classURI"));
```

```
    theFeatures.put("inst", theFeatures.get("URI"));
```

do the same
for inst

```
  }
```

```
}
```

Hands-on 7: ontology-aware JAPE

- Load the JAPE transducer *rename-lookup-features.jape* and add to the end of your existing pipeline
 - Set the input and output sets for it to *Test*
- Run the modified pipeline to see how some of the Lookup annotations in Test now have class features
- Load the JAPE transducer *person-onto-matching.jape* and add it to the end of the pipeline as before.
 - Set the input and output sets for it to *Test*
 - Give the ontology as the run-time param
- Run the modified pipeline to see how it creates new *Mention* annotations
- Save the application with a new name and close it

Extra Hands-on

- Modify the ontology:
 - Add a subclass Country of the Location class
 - Add an instance of Country called **UK**
- ***You will need to recreate the OntoRootGazetteer if you modify the ontology***
- In a text editor, open *person-onto-matching.jape* and add a similar rule that matches any location and creates a Mention annotation
- Reinitialise the JAPE grammar in GATE
- Run the application: it should now annotate UK as a Mention too

Semantic Annotation in the real world

- Usually, gazetteer-based annotation isn't enough for real applications
- You can take the ANNIE results and map them to ontology classes using JAPE to create Mention annotations
 - Organization → Mention.class=Organization
- Rules can also combine Lookups from traditional gazetteers with Lookups from ontologies and other clues, in order to detect Mentions
- Co-referring items may need linking to the ontology
 - e.g. Mr. Brown = Gordon Brown = he
- Disambiguation: if several instances with label “John Smith”, pick the correct one
 - Context from the text can be matched against the ontology

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 ?cls
```

```
WHERE {
```

```
    # Country
```

```
    UNION {
```

```
        ?inst rdf:type dbp:Country .
```

```
        ?inst foaf:name ?label .
```

```
        FILTER (lang(?label) = "en")
```

```
        ?inst a ?cls .
```

```
        ?cls a owl:Class .
```

```
        FILTER (?cls = dbp:Country)
```

```
    }
```

```
}
```

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. The current version is likely to change significantly in the near future
- 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 the demonstrations on Thu, the add-on track on Fri & <http://gate.ac.uk/userguide/sec:gazetteers:lkb-gazetteer>

Performance Evaluation

- Mention annotations can be evaluated against a gold standard by matching the classes or instances
- However, traditional IE evaluation measures (Precision and Recall) don't take into account the class hierarchy
- Some mistakes can be “more wrong” than others
 - Nick Clegg → Person (not Leader) – still logically correct
 - Nick Clegg → Location – wrong
- We need a way of dealing with this, to give some credit for these kind of situations

Balanced Distance Metric

- BDM measures the closeness of two concepts in an ontology or taxonomy
- It produces a real number between 0 and 1
- The more closely related the two concepts are in an ontology, the greater their BDM score is
- It is dependent on a number of features:
 - the length of the shortest path connecting the two concepts
 - the depth of the two concepts in the ontology
 - size and density of the ontology

Ontology-sensitive F-measure

- The IAA plugin computes precision, recall, and F-measure over a corpus
- It can be set to calculate ontology-sensitive F-measure using the BDM score to replace the number of correct matches
- More information about BDM in

D. Maynard, Y. Li and W. Peters. NLP Techniques for Term Extraction and Ontology Population (2007)

<http://gate.ac.uk/gate/doc/papers.html>

Use OAT to create gold standard

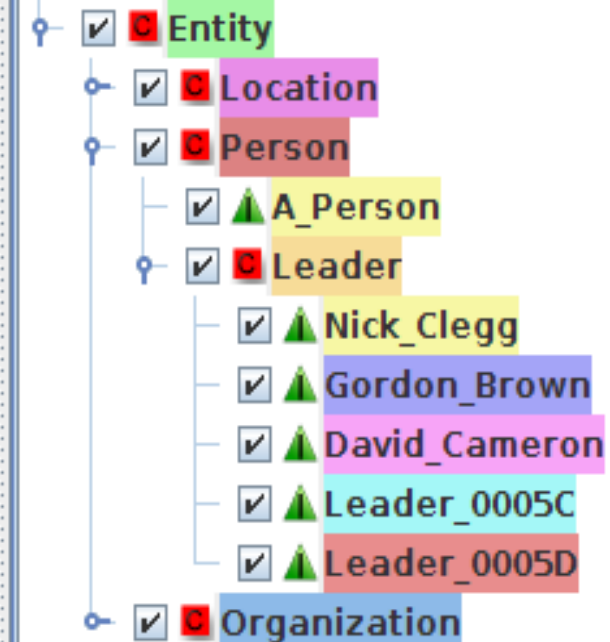
David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

Labour leader Gordon Brown went to vote shortly after 1100 BST at a community centre close to his home in North Queensferry, Fife. His wife Sarah was with him.

Nick Clegg, leader of the Liberal Democrats, arrived at a polling station in Sheffield Hallam at 1120 BST. His wife Miriam is unable to vote in the general election because she is a Spanish citizen.

The leader of the Scottish National Party, Alex Salmond, cast his vote shortly before noon, at Macduff in Banffshire. Ieuan Wyn Jones of Plaid Cymru voted in the constituency of Ynys Mon in north Wales at lunchtime.

test-ontology-instances.owl_0227



By convention, change the OAT default to put the annotations in the Key set. It is already configured to create Mentions with class and inst features.



Traditional Precision/Recall

Annotation Diff Tool

Key doc: voting-example.xml...
Resp. doc: voting-example.xml...

Key set: Key
Resp. set: result

Type: Mention
Features: ☐ all ☒ some ☐ none

Weight: 1.0

Compare

Key	Features	=?	Start	End	Response	Features
David·Cameron	{ontology=http://gat...ample#David_Cameron}	=	672	685	David·Cameron	{class=http://gate.a...ample#David_Camer
Gordon·Brown	{ontology=http://gat...ample#Gordon_Brown}	=	887	899	Gordon·Brown	{class=http://gate.a...ample#Gordon_Bro
Alex·Salmond	{ontology=http://gat...ample#Leader_0005C}	-?				
Ieuan·Wyn·Jones	{ontology=http://gat...ample#Leader_0005D}	-?				
Nick·Clegg	{ontology=http://gat.../example#Nick_Clegg}	<>	1034	1044	Nick·Clegg	{class=http://gate.ac.uk/example#Person}

Correct: 2
Partially correct: 0
Missing: 3
False positives: 1

Recall: 0.40
Precision: 0.67
F-measure: 0.50

Statistics

Adjudication

1 documents loaded

Show document

Export to HTML

BDM PR in GATE

- Located in the Ontology_BDM_Computation plugin
- Requires an ontology (as a file) to compute and outputs the results in a file
- For each pair of classes in the ontology, it calculates a number of statistics
- Since BDM is symmetric for any two concepts, the resulting file contains only one entry per pair, despite one being called key
- Example file: **bdm-output.txt** (do not overwrite this!)

```
key=http://gate.ac.uk/example#Entity,  
response=http://gate.ac.uk/example#Location, bdm=0.0,  
msca=http://gate.ac.uk/example#Entity, cp=0, dpk=0, dpr=1, n0=1.6666666,  
n1=1.6666666, n2=2.0, bran=1.8000001
```

Hands-on 8: Corpus QA with BDM

- Load the document `voting-example-bdm.xml` and add it to a corpus
- Click on the *Corpus Quality Assurance* tab at the bottom of the corpus view
- In *Annotation Sets*, select *Key* as A (Key) and *Test* as B (Response)
- Under *Annotation Types*, select *Mention*
- Highlight *Mention* and then select class under *Annotation Features*
- Click on *F-score* under *Measures*, and select *F1.0-score average BDM*
- Click on *Options* in the selection pane and select the file *bdm-output* as the value of the BDM file
- Click *Compare* and view the results

Our example text again

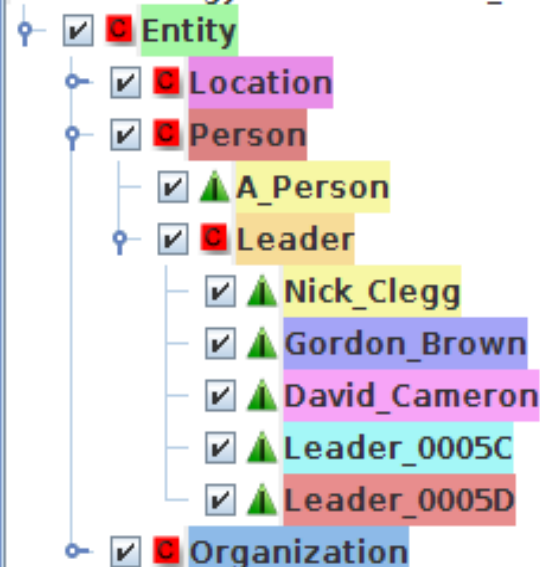
David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

Labour leader Gordon Brown went to vote shortly after 1100 BST at a community centre close to his home in North Queensferry, Fife. His wife Sarah was with him.

Nick Clegg, leader of the Liberal Democrats, arrived at a polling station in Sheffield Hallam at 1120 BST. His wife Miriam is unable to vote in the general election because she is a Spanish citizen.

The leader of the Scottish National Party, Alex Salmond, cast his vote shortly before noon, at Macduff in Banffshire. Ieuan Wyn Jones of Plaid Cymru voted in the constituency of Ynys Mon in north Wales at lunchtime.

test-ontology-instances.owl_022:



Clegg is marked as a Person, instead of Leader

Salmond is missing

Results

- The traditional scores:
 - Match = 2, Only A (missing) = 2, Only B (spurious) = 1, Overlap (Partial) = 0
 - Recall = 0.50, Precision = 0.67, F1 = 0.57
- BDM-sensitive scores:
 - Recall = 0.60, Precision = 0.8, F1 = 0.69
(The BDM Match = 2.418, not shown)



Summary

- Brief introduction to ontologies, semantic annotation
- Manual ontology editing in GATE
- Manual semantic annotation using OAT and RAT
- Automatic semantic annotation using OntoRootGazetteer and ontology-aware JAPE
- Ontology-based evaluation using BDM

Further materials

Ontology design principles:

<http://lsdis.cs.uga.edu/SemWebCourse/OntologyDesign.ppt>

BDM:

<http://gate.ac.uk/userguide/sec:eval:bdmplugin>

Semantic Annotation:

- K. Bontcheva, B. Davis, A. Funk, Y. Li and T. Wang. Human Language Technologies. Semantic Knowledge Management, John Davies, Marko Grobelnik, and Dunja Mladenic (Eds.), Springer, 37-49, 2009.
- K. Bontcheva, H. Cunningham, A. Kiryakov and V. Tablan. Semantic Annotation and Human Language Technology. Semantic Web Technology: Trends and Research. John Wiley and Sons Ltd. 2006.
- D. Maynard, Y. Li and W. Peters. NLP Techniques for Term Extraction and Ontology Population. Bridging the Gap between Text and Knowledge - Selected Contributions to Ontology Learning and Population from Text, P. Buitelaar and P. Cimiano (editors). IOS Press, 2007.