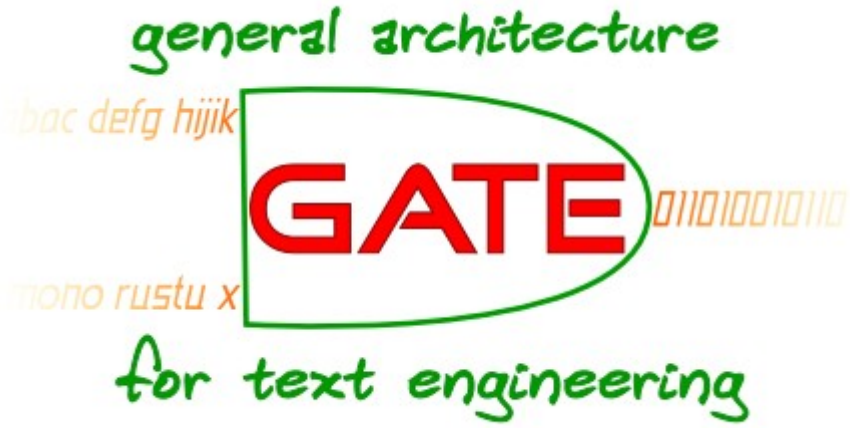


Ontologies, semantic annotation and GATE

Kalina Bontcheva
Johann Petrak
University of Sheffield

Topics

- Ontologies
- Semantic annotation
- Ontology population
- Ontology learning

Ontology - What?

- “An Ontology is a formal specification of a shared conceptualisation.” [Gruber]
- Set of concepts (instances and classes)
- Relationships between concepts (is-a, is-subclass, is-part, located-in)
- Allows reasoning
  - Class membership, inferred properties ...
  - Need tradeoff: expressivity vs. reasoning complexity and decidability

Ontology – How?

- **RDF/RDFS** – Triple-based representation scheme
- **OWL 1.1 / OWL 2** – Ontology representation formalism based on RDF/RDFS
- **Description Logic** – Logic based KR formalism used for OWL, allows well-defined sublanguages.
- OWL 1.1: OWL-Lite, OWL-DL, OWL-Full official sublanguages, several inofficial others
- OWL 2: language profiles

  ==> expressiveness / reasoning effort trade-off

OWL – Issues

- **OWA** – Open World Assumption: if something is not in the ontology, it can still be true
- **No UNA –** No Unique Name Assumption: one entity can have different names
- **owl:Class** vs. rdfs:Class

Ontologies in GATE

- Abstract ontology model for the API:
- Comes with one concrete implementation preinstalled: Sesame/OWLIM
- Comes with several tools:
  - Ontology Visualizer/Editor
  - OntoGazetteer, OntoRootGazetteer
  - Ontology support in JAPE

Ontology implementation

- SwiftOWLIM2 from Ontotext
- A Sesame1 repository SAIL
- Fast in memory repository, scales to millions of statements (depending on RAM)
- Supports “almost OWL-Lite”
- SwiftOWLIM is exchangeable with persistence-based BigOWLIM: not free, scales to billions of statements.
- Planned: Migration to Sesame2/OWLIM3

Ontology API

- Ontology, Ontology resources represented as Java objects: `gate.creole.ontology`
- Ontology, OClass, OResource, URI, Literal
- Currently: ~ OWL-Lite actions
- OWLIMOntologyLR is a Java Ontology object
- JAPE RHS can access Ontology object

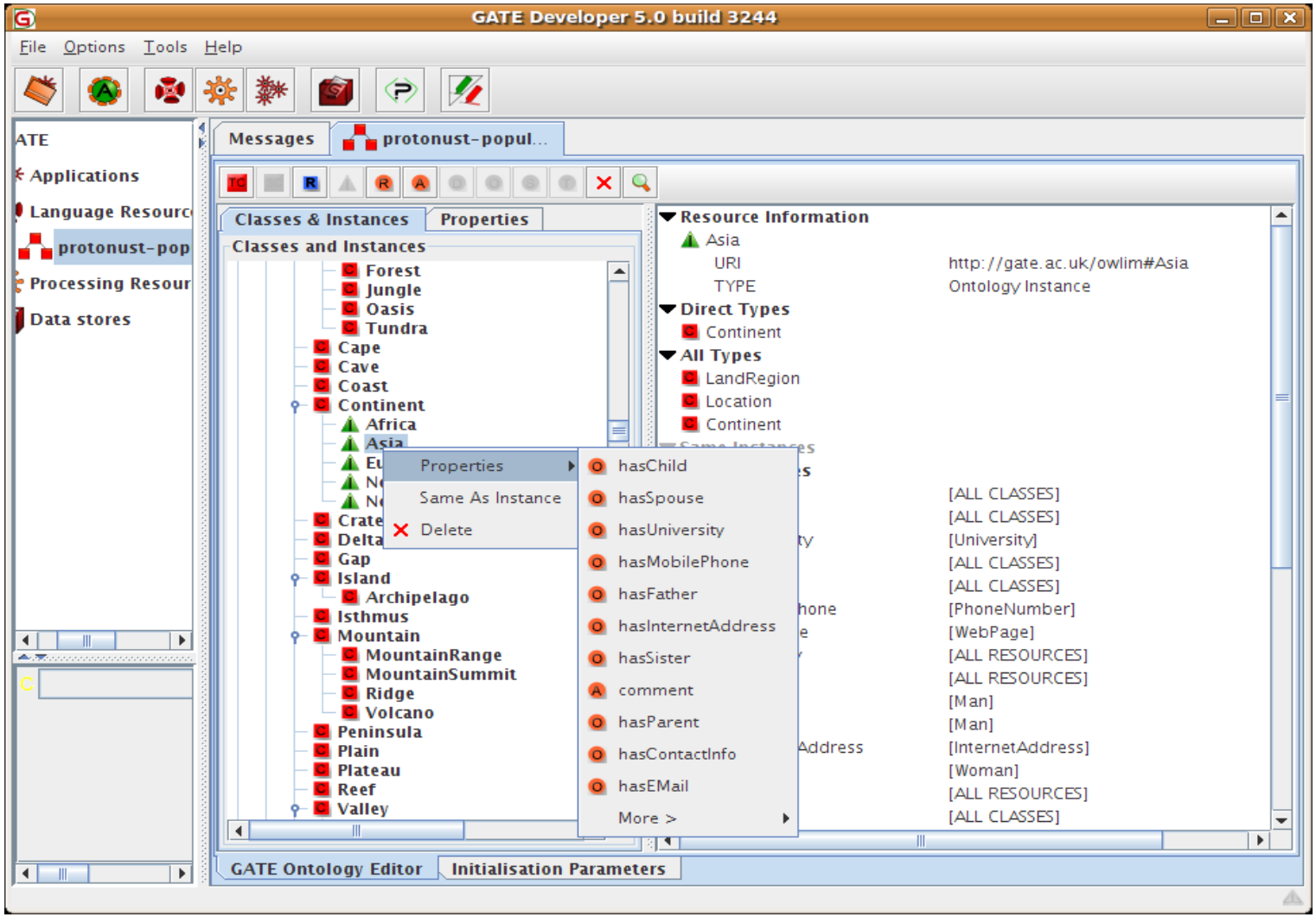
Ontology API


```
URI uri = new URI("http://my.uri/#Class1",false);

OClass c = ontology.addClass(uri);

Datatype dt = new Datatype(XMLStringURI);

DatatypeProperty dtp =
   ontology.addDatatypeProperty(uri2,domain,dt);

OInstance i = ontology.addOInstance(uri3,c);

Set<OClass> scs =
  c.getSuperClasses(DIRECT_CLOSURE);

i.addDatatypePropertyValue(dtp,
  new Literal("thevalue"));
```

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

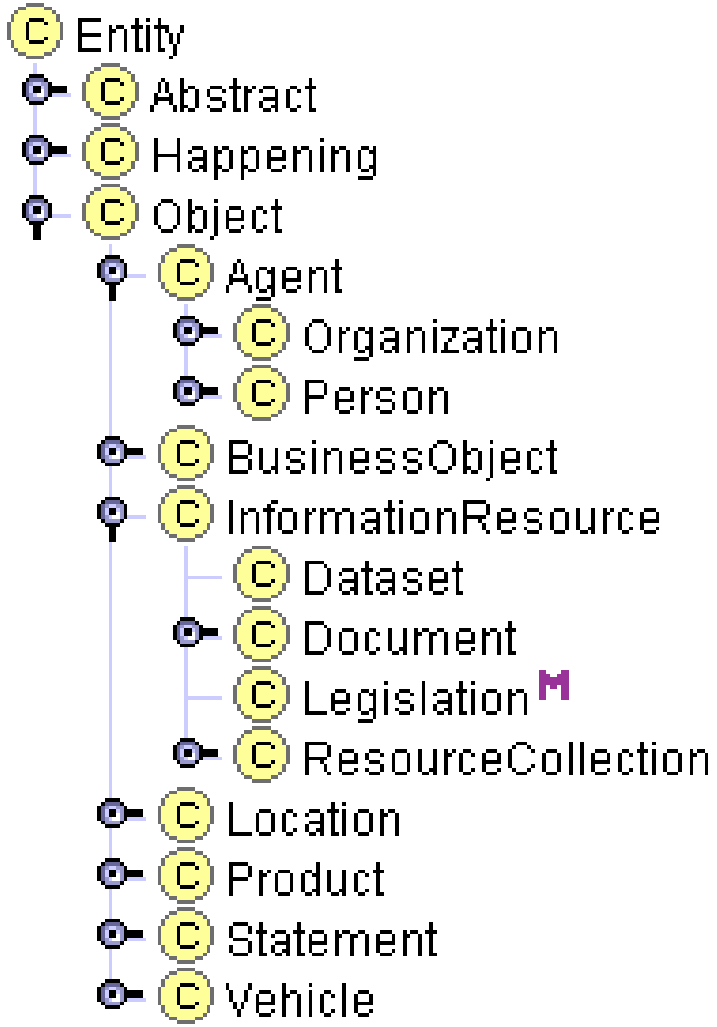
Ontology Editor



PROTON Ontology

- **a light-weight upper-level ontology**;
- **250 NE classes**;
- **100 relations and attributes**;
- **200.000 entity descriptions;**
- covers mostly **NE classes**, and ignores general concepts;
- includes classes representing **lexical resources**.

proton.semanticweb.org



Hands-on 1

- Load Ontology_Tools plugin
- Language Resource → New → OWLIMOntologyLR
  - URI: load from web or from local file: load protonust.owl
  - Format: rdfxml, ntriples, turtle
  - Default default NS: http://gate.ac.uk/owlim#
- Resolves all imports automatically when loading
- Double-click ontology LR to view/edit

Semantic Annotation

- “Semantic”: link the annotation to a concept in an ontology.
- The semantic link connects the text mention to knowledge about the concept that is mentioned.
- The mention can link to an instance, a class, or a property – i.e. to a resource
- Use the semantic link to access additional data about the concept – use for disambiguation and further annotation processing
- Use for NER, IE, querying, ...

Semantic Annotation

Ontology

```
:London a City ;
  ...
:Company a :Organization .

XYZ-02FA a :Company ;
  rdfs:label “XYZ”@en ;
  :basedIn :London-UK
  ...
XYZ-98 a :Company ;
  rdfs:label “XYZ”@en ;
  :basedIn :Boston-US
  …
```

Document

XYZ was established on 03 November 1978 in London. The company opened a plant in Bulgaria in ..

Semantic Annotation

Ontology

```
:London a City ;
  ...
:Company a :Organization .

XYZ-02FA a :Company ;
  rdfs:label “XYZ”@en ;
  :basedIn :London-UK
  ...
XYZ-98 a :Company ;
  rdfs:label “XYZ”@en ;
  :basedIn :Boston-US
  …
```

Document

XYZ was established on 03 November 1978 in London. The company opened a plant in Bulgaria in ..

Semantic Annotation vs. "traditional"

- Link to hierarchy of concepts instead of flat set of concepts
- Larger space of possible annotations
- - harder to get it right
- + candidate concepts have associated knowledge that can be used to support decision
- + found concepts can be generalized based on ontology: context(company) < context(organization)
- → ontology aware JAPE in GATE

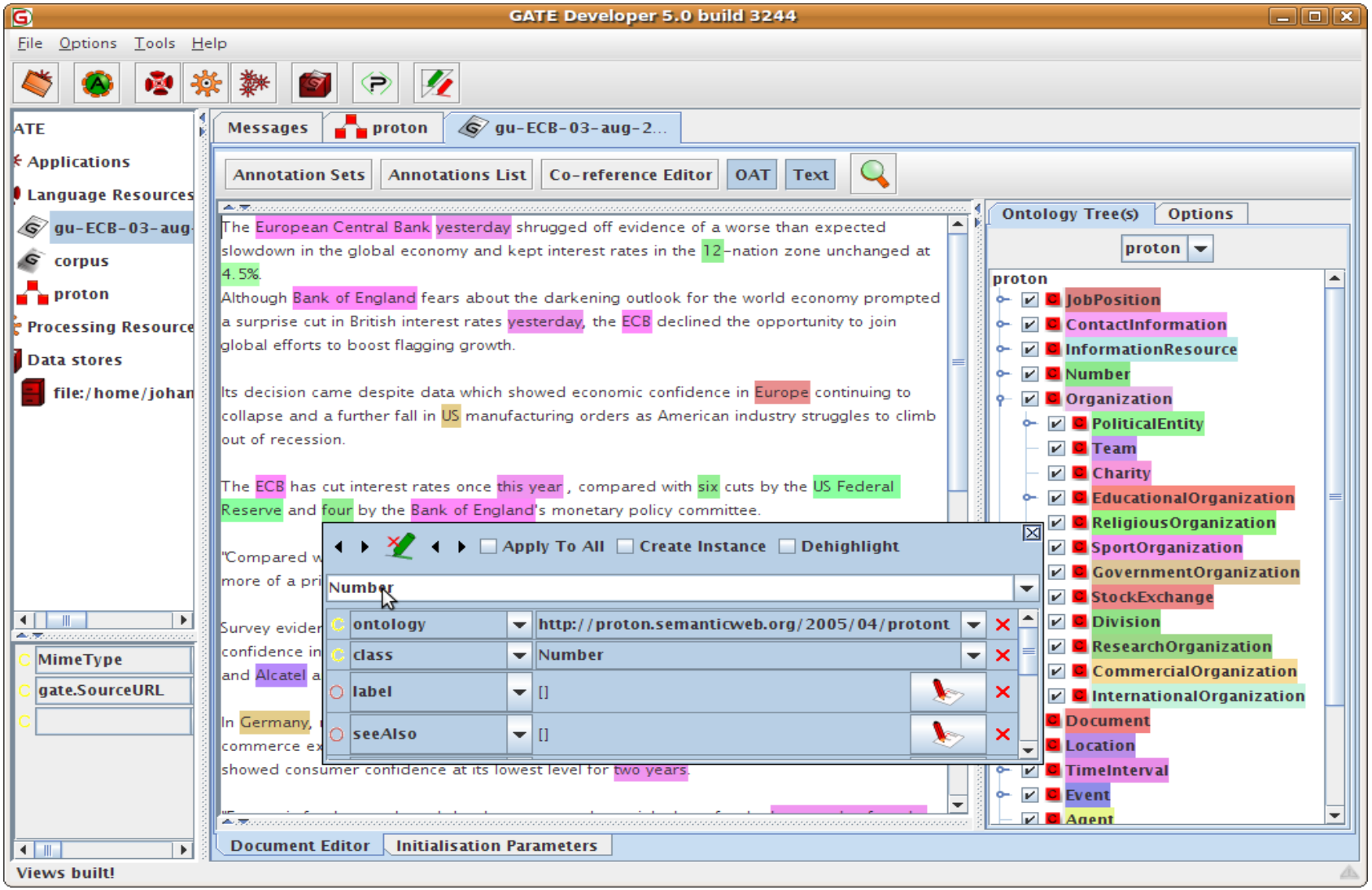
Semantic Annotation: How?

- Manually: ontology based annotation – GATE OAT (Ontology Annotation Tool)
- Automatically
  - Gazetteer/rule/pattern based
  - Similarity based
  - Classifier (ML) based
  - Parser based
  - Combinations thereof

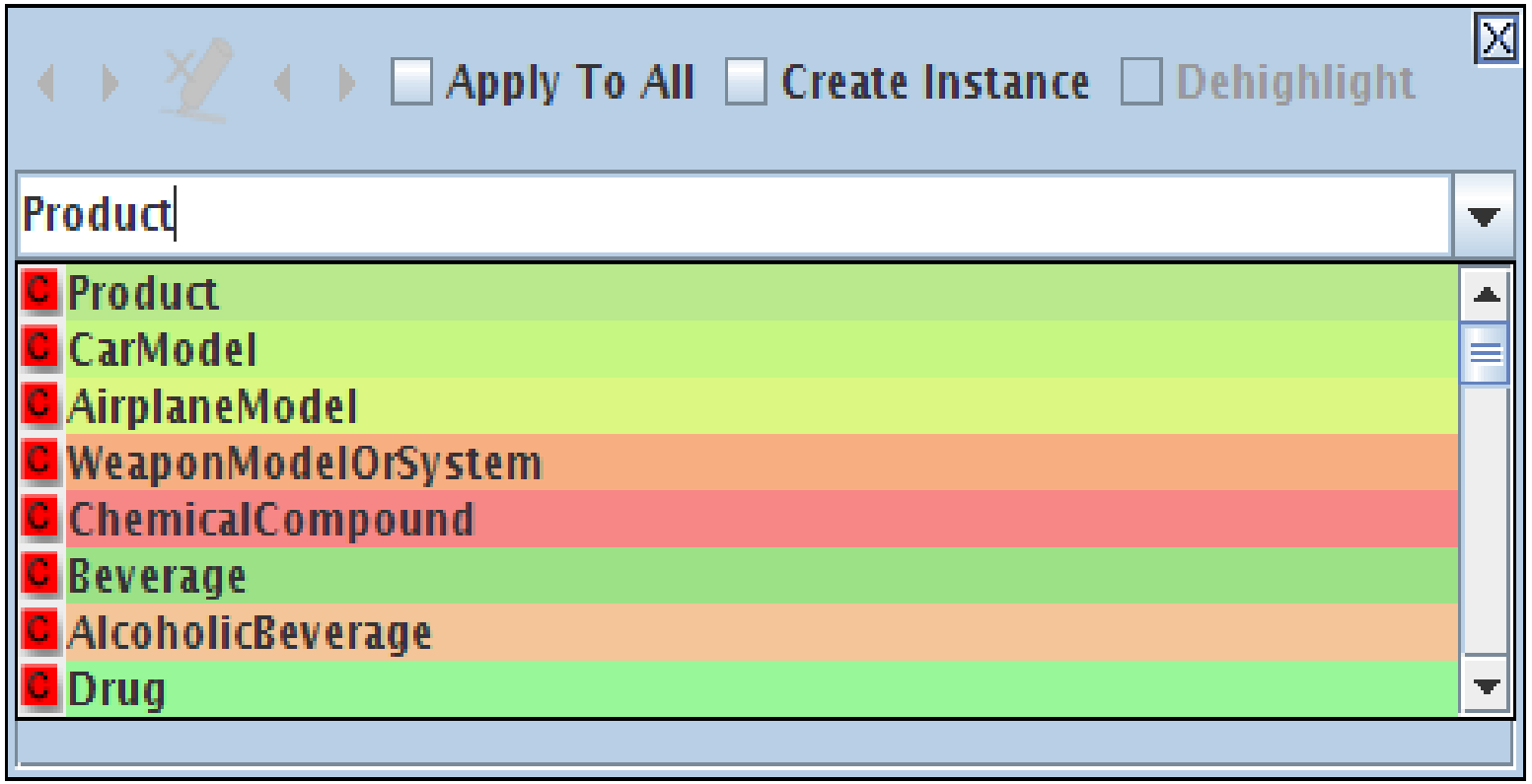
GATE OAT

- Show 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
- For:
  - Creating Evaluation Corpus
  - Creating ML-Training Corpus

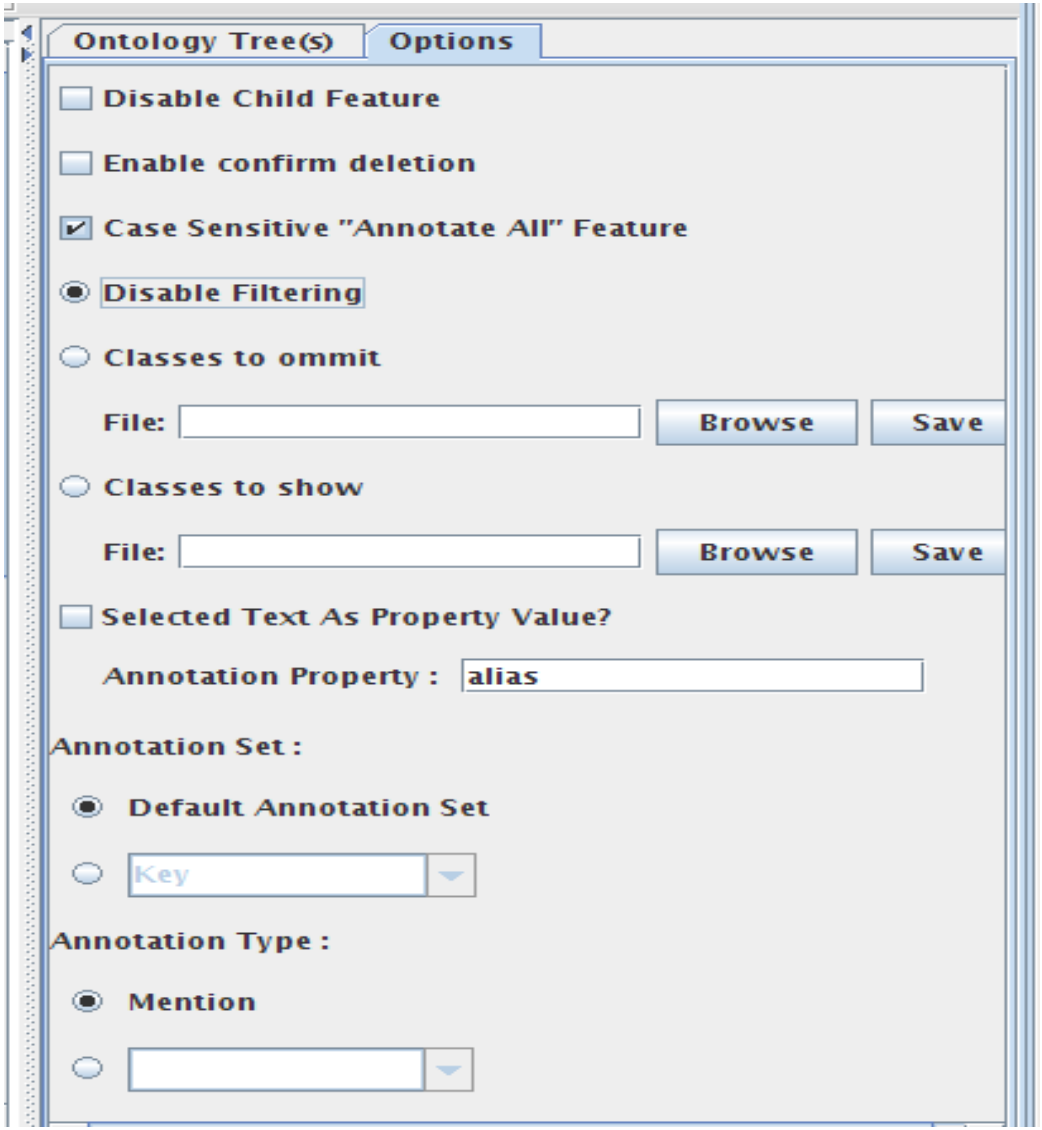
OAT


GATE Developer 5.0 build 3244
File Options Tools Help
ATE
Applications
Language Resources
gu-ECB-03-aug
corpus
proton
Processing Resource
Data stores
file:/home/johan
MimeType
gate.SourceURL
Messages
proton
gu-ECB-03-aug-2...
Annotation Sets
Annotations List
Co-reference Editor
OAT
Text
The European Central Bank yesterday shrugged off evidence of a worse than expected slowdown in the global economy and kept interest rates in the 12-nation zone unchanged at 4.5%.
Although Bank of England fears about the darkening outlook for the world economy prompted a surprise cut in British interest rates yesterday, the ECB declined the opportunity to join global efforts to boost flagging growth.
Its decision came despite data which showed economic confidence in Europe continuing to collapse and a further fall in US manufacturing orders as American industry struggles to climb out of recession.
The ECB has cut interest rates once this year , compared with six cuts by the US Federal Reserve and four by the Bank of England's monetary policy committee.
Apply To All
Create Instance
Dehighlight
Number
ontology
http://proton.semanticweb.org/2005/04/protont
class
Number
label
[]
seeAlso
[]
showed consumer confidence at its lowest level for two years.
Ontology Tree(s)
Options
proton
proton
JobPosition
ContactInformation
InformationResource
Number
Organization
PoliticalEntity
Team
Charity
EducationalOrganization
ReligiousOrganization
SportOrganization
GovernmentOrganization
StockExchange
Division
ResearchOrganization
CommercialOrganization
InternationalOrganization
Document
Location
TimeInterval
Event
Agent
Document Editor
Initialisation Parameters
Views built!

OAT


Apply To All
Create Instance
Dehighlight
Product
Product
CarModel
AirplaneModel
WeaponModelOrSystem
ChemicalCompound
Beverage
AlcoholicBeverage
Drug

OAT


Ontology Tree(s)
Options
Disable Child Feature
Enable confirm deletion
Case Sensitive "Annotate All" Feature
Disable Filtering
Classes to ommit
File:
Browse
Save
Classes to show
File:
Browse
Save
Selected Text As Property Value?
Annotation Property :
alias
Annotation Set :
Default Annotation Set
Key
Annotation Type :
Mention

Hands-on 2

- (Load Ontology_Tools plugin)
- Load ontology protonust.owl
- Load a document from `corpus_original` (encoding iso-8859-1)
- Create annotation
- Create annotation and instance
- Load document from `corpus_annotated` and show annotations

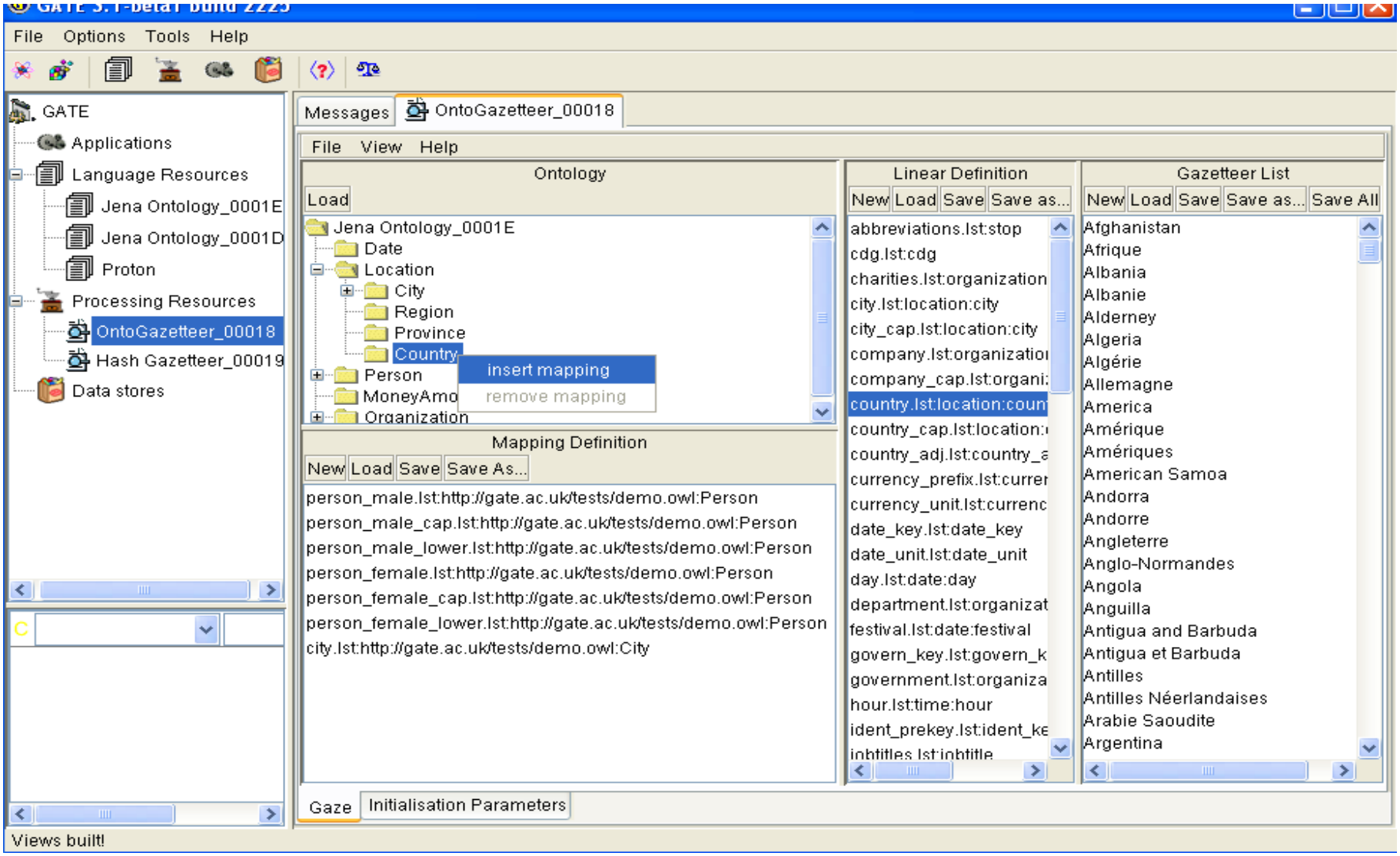
Semantic Annotation: Automatic

- Create language resources from existing ontology:
  - Retrieve or generate possible mentions and create gazetteer lists or gazetteer
  - Preprocess document
  - Annotate document with gazetteer
  - Disambiguation, postprocessing

OntoGazetteer

- Map ontology classes to gazetteer lists
- e.g. List of first names to class “Person”
- Uses Hash Gazetteer internally
- Provides a GUI to establish the mappings
- Mapping file could also be created by other means
  - Gazetteer list file name / ontology class URI
- For simple situations w/ few classes and many instances per class

OntoGazetteer



Onto Root Gazetteer

- Tries to find mentions in resource names (fragement ids), data property values, labels
- Converts “CamelCase” names, hyphen, underscore
- Produce multiword subsequences
- Finds lemma of mentions using the GATE Morphological Analyzer
- Creates a gazetteer PR that can be used with the FlexibleGazetteerPR

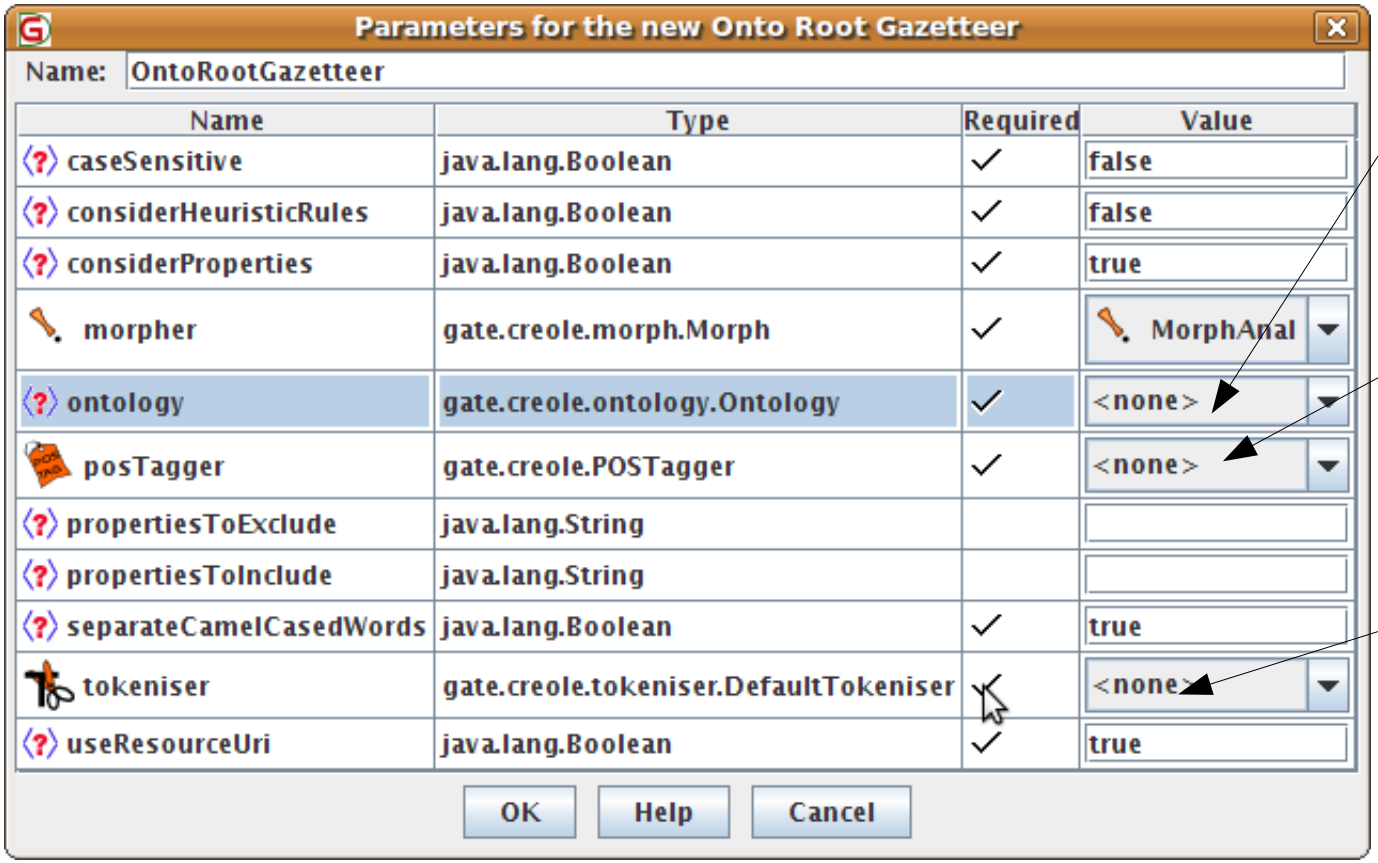
Onto Root Gazetteer

- OntoRootGazeteer:
  - Generate candidate list from ontology
  - Run Tokeniser, POS tagger, Morphological Analyser(M.A.) and find lemmata/stems
- Document pipeline:
  - Run Tokenizer, POS tagger, M.A. and find lemmata/stems and place in Token.root
- Flexible gazetteer:
  - Match Token.root (*not text* as DefaultGazetteer) using OntoRootGazetteer

Hands-on 3

- Plugin Ontology_Tools for OntoRootGazetteer
- Plugin Tools for GATE Morphological Analyser
- Load Ontology
- Create Tokeniser, POS Tagger, and Morphological Analyser
- Create and configure OntoRootGazetteer
- Create Flexible Gazetteer
  - add OntoRootGazetteer as gazetteerInst
  - Specify Token.root for inputFeatureNames

Hands-on 3



Ontology LR

POS Tagger PR

Tokeniser PR

Hands-on 3

- Create pipeline
- Create and add Sentence splitter
- Add Tokeniser
- Add POS Tagger
- Add Morphological Analyser
- Add Flexible Gazetteer
- Run

Postprocess

- Original annotations contain just candidate URIs and classes.
- Original annotations might overlap
- Pull in additional knowledge for
  - Disambiguation (which person of that name?)
  - Semantic enrichment for subsequent processing stages

Ontology-aware JAPE

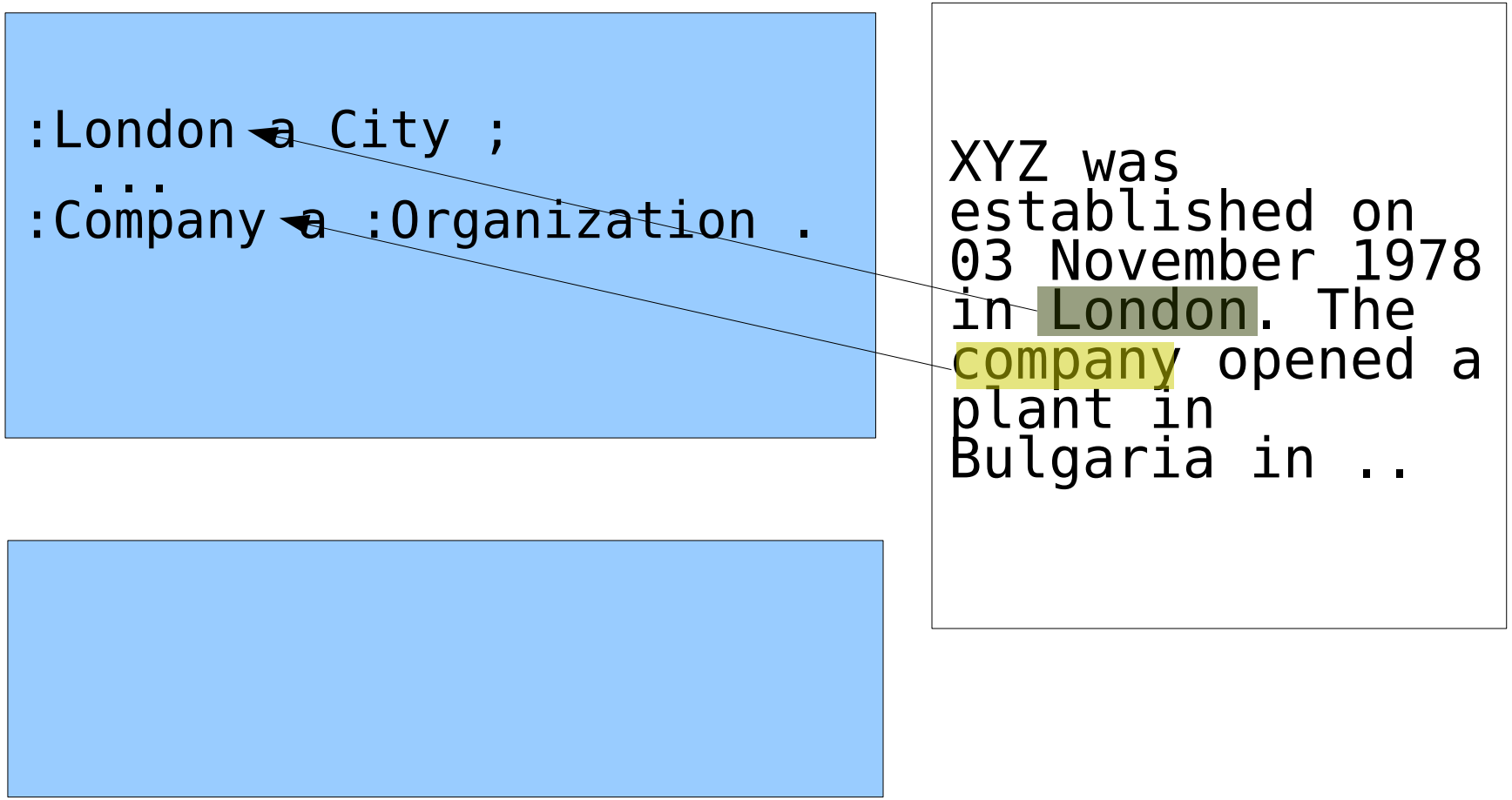

```
Rule: LocationLookup
(
  {Lookup.class == Location}
):location
-->
:location.Location = { }
```

Matches any name of a class that is a subclass of Location

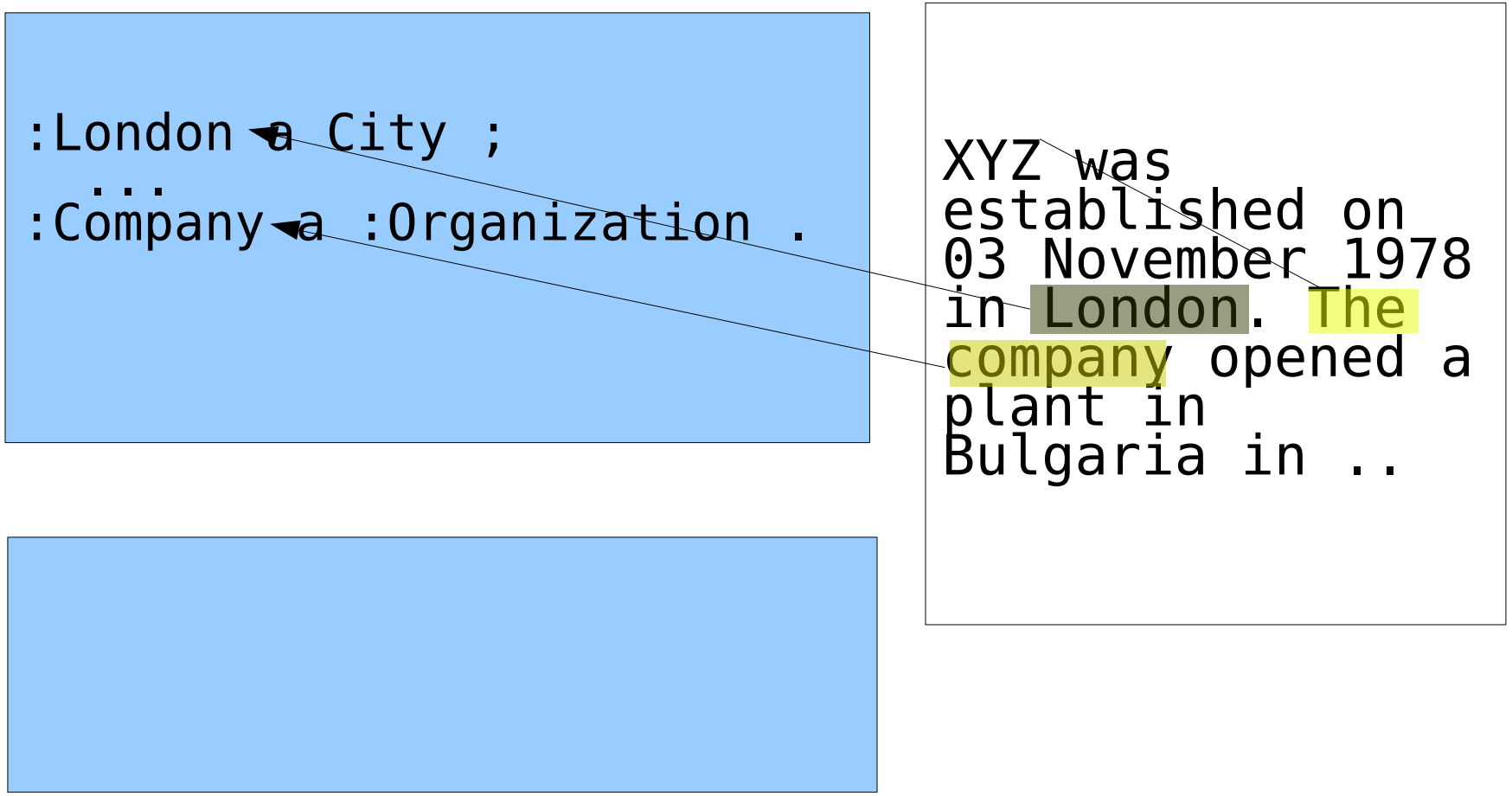
Ontology Population

- Annotate document and find mentions of what could be (new) instances in the ontology
  - Use traditional NER, linked to ontology
  - Use semantic annotation based on existing knowledge
  - Use ML
- Create ontology instances and property values ("ABOX") from the final annotations

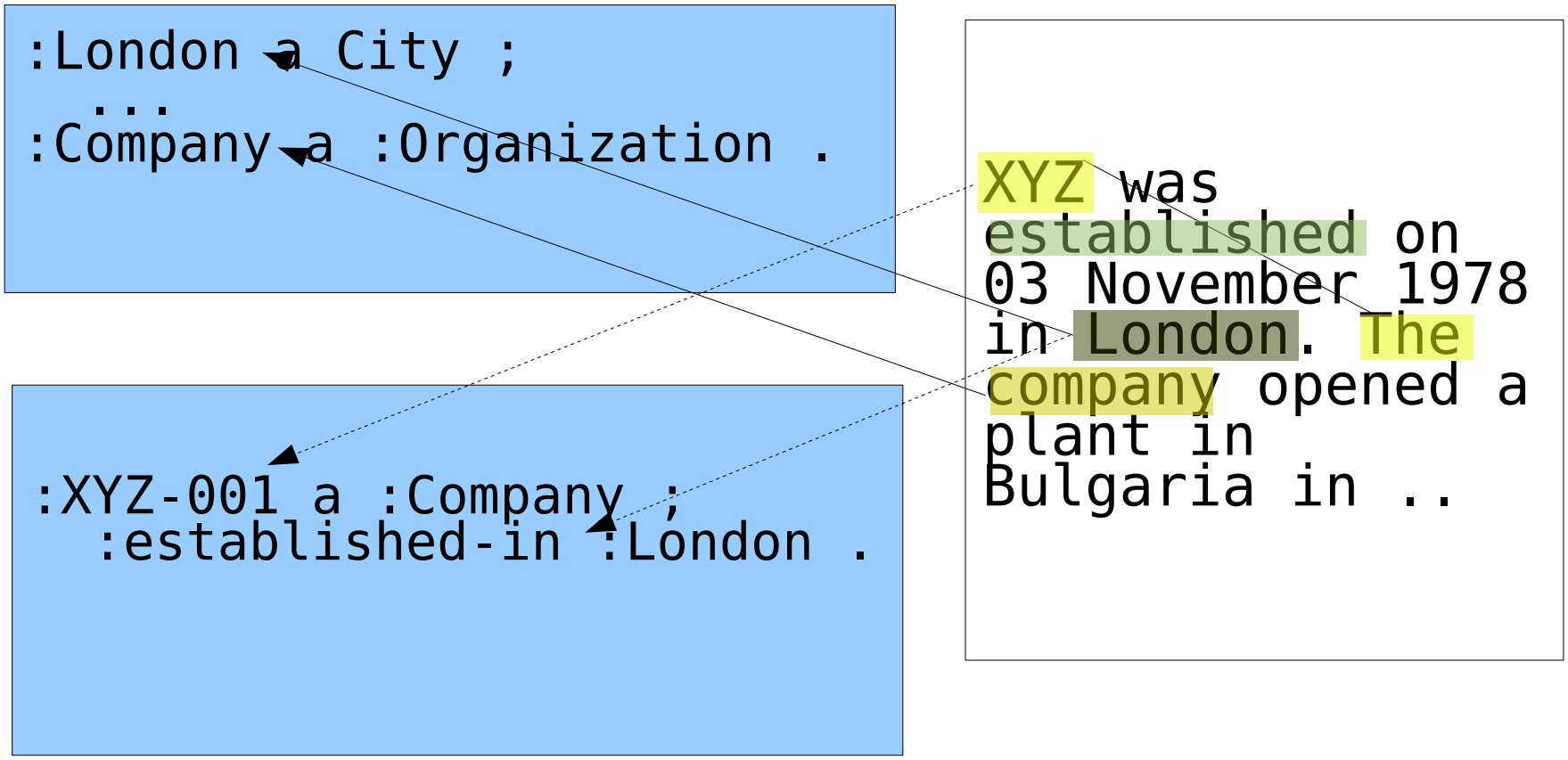
Ontology population



Ontology population



Ontology population



Ontology Population

- Populate Ontology with Instances:
  - Of classes
  - Of properties connecting class instances with other class instances or values (literals)
  - Graph describing n-ary relations or events …
- Strategy
  - Place in domain ontology?
  - Place in intermediate ontology/KB?

Ontology Population

- Place directly in domain ontology:
  + Simple & straight-forward

  - Cannot model likelihoods, hard to model meta information (where from, which context)

  Can easily leave sub-language or become inconsistent

  Knowledge arrives incrementally but has dependencies
- Place in intermediate ontology

  - Processing more complex

  Apropriate model for intermediate ontology?

  + Can do iterative improvement

  Can model meta information

Ontology Population: JAPE

```
Rule: FindEntities
({Mention}):mention
–>
:mention{
  Annotation mentionAnn =
   (Annotation)mentionAnnots
     .iterator().next();

  String className =
   (String)mentionAnn
     .getFeature().get("class");

  List<OResource> matches =
    ontology.getOresourcesByName(className);
```

Use qualified name!

Check if null!

Ontology Population: JAPE

```
// find the resource representing the class
for(OResource aResource : matches ) {
  if(aResource instanceof Oclass) {
    aClass = (Oclass) aResource;
    Break;
  }
}

// get Text of mention
String mentionName =
  doc.getContent().
    getContent()(
      mentionAnn.getStartNode().getOffset(),
      mentionAnn.getEndNode().getOffset()).
    toString();
```

Ontology Population: JAPE

```
// populate the ontology
gate.creole.ontolog.URI uri =
  OntologyUtilities.createURI(
    Ontology, mentionName, false);

if(!ontology.containsOInstance(uri)) {
  ontology.addOInstance(uri, aClass);
}
```

Hands-on 4

- Open protonust.owl ontology
- Create corpus from corpus_annotated (encoding iso-8859-1)
- Create JAPE file populate.jape or download populate.jape from http://gate.ac.uk/wiki/Upload.jsp?page=FIG09
- Create Pipeline and run JAPE transducer
- View ontology

Recap

- **Semantic Annotation**
  - Mentions of instances in the text are annotated wrt concepts (classes) in the ontology.
  - Requires that instances are disambiguated.
  - It is the **text** 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.
  - It is the **ontology** which is modified.

Ontology Learning

- **Extraction** of (domain) ontologies from natural language text
  - Machine learning
  - Natural language processing

- **Tools:** OntoLearn, OntoLT, ASIUM, Mo'K Workbench, JATKE, TextToOnto, …

Slide courtesy of Johanna Volker, UKARL

Ontology Learning – Tasks

Concept extraction	`car, vehicle, person`
Concept classification	`subclass-of( car, vehicle )`
Instance extraction	`Peter, his-car`
Instance classification	`instance-of( Peter, person )`
Relation extraction	`drive( person, car )`
Relation instance extraction	`drive( Peter, his-car )`

Slide courtesy of Johanna Volker, UKARL

OL – Problems
Text Understanding

- Words are ambiguous
  - 'A bank is a financial institution. A bank is a piece of furniture.'

→ `subclass-of( bank, financial institution )` ?

- Natural Language is informal
  - 'The sea is water.'

→ `subclass-of( sea, water )` ?

- Sentences may be underspecified
  - 'Mary started the book.'

→ `read( Mary, book_1 )` ?

- Anaphores
  - 'Peter lives in Munich. This is a city in Bavaria.'

  → `instance-of( Munich, city )` ?

- Metaphores, …

Slide courtesy of Johanna Volker, UKARL

OL – Problems Knowledge Modeling

- What is an instance / concept?
  - 'The koala is an animal living in Australia.'

  → `instance-of( koala, animal )`

  `subclass-of( koala, animal )` ?
- How to deal with opinions and quoted speech?
  - 'Tom thinks that Peter loves Mary.'

  → `love( Peter, Mary )` ?
- Knowledge is changing
  - `instance-of( Pluto, planet )` ?

Conclusion:
- Ontology learning is difficult.
- What we can learn is fuzzy and uncertain.
- Ontology maintenance is important.

Slide courtesy of Johanna Volker, UKARL

Ontology Learning Approaches
Concept Classification

- Heuristics
  - 'image processing software'

  → `subclass-of( image processing software, software )`
- Patterns
  - 'animals such as dogs'
  - 'dogs and other animals'
  - 'a dog is an animal'

→ `subclass-of( dog, animal )`

Slide courtesy of Johanna Volker, UKARL

JAPE Patterns for Ontology Learning


```
rule: Hearst_1
(
   (NounPhrase):superconcept

   {Token.string=="such"}

   {Token.string=="as"}

   (NounPhrasesAlternatives):subconcept
):hearst1
-->
:hearst1.SubclassOfRelation = { rule = "Hearst1" },
:subconcept.Domain = { rule = "Hearst1" },
:superconcept.Range = { rule = "Hearst1" }
```

Slide courtesy of Johanna Volker, UKARL