

---

# Ontologies and semantic annotation

---

---

# Ontology – A Definition



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



# What is an Ontology?

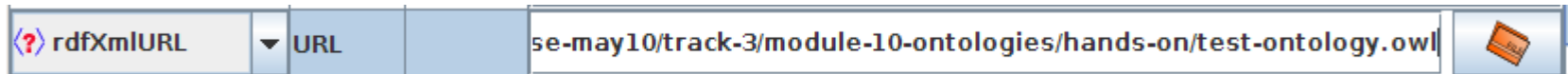
- Set of concepts (instances and classes)
  - Relationships between them (is-a, part-of, located-in)
  - Multiple inheritance
    - Classes can have more than one parent
    - Instances can have more than one class
  - Ontologies are graphs, not trees
- ▼ **C** Company
    - C** Airline
    - C** Bank
    - C** InsuranceCompany
    - ▼ **C** MediaCompany
      - C** NewsAgency
      - C** PublishingCompany
      - C** TVCompany
    - C** PublicCompany
    - ▼ **C** SportClub
      - C** SoccerClub
    - C** Telecom



# 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

The screenshot displays the GATE Developer 5.0 build 3244 interface. The main window is titled "GATE Developer 5.0 build 3244" and contains several panes:

- Left Pane:** A navigation tree with categories like "Applications", "Language Resources", "Processing Resources", and "Data stores". The "protonust-popul..." ontology is selected.
- Classes & Instances Pane:** A tree view showing the ontology's structure. The "Asia" class is selected, and a context menu is open over it. The menu options include "Properties", "Same As Instance", "Delete", "hasChild", "hasSpouse", "hasUniversity", "hasMobilePhone", "hasFather", "hasInternetAddress", "hasSister", "comment", "hasParent", "hasContactInfo", "hasEMail", and "More >".
- Resource Information Pane:** Displays details for the selected "Asia" instance, including its URI (<http://gate.ac.uk/owlim#Asia>), type ("Ontology Instance"), and a list of "Direct Types" and "All Types" (both containing "Continent").



# URIs, Labels, Comments

---

- The names of classes and instances shown in the editor is what is called a URI
  - `http://gate.ac.uk/example#Person`
  - URIs cannot have spaces (and other such characters)
- 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**





# Hands-on 1

- 
- 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 few subclasses of Locations, e.g., countries
  - Add yourself as an instance of the class Person
  - Add a label with your full name
  - Keep it open for the next hands on



# Datatype Properties

---

- Datatype properties link individuals to data values
- Datatype properties can be of type boolean, date, int, ...
  - E.g., people can have an age property
  - Available datatypes taken from XMLSchema
- To define a new data property
  - Click on the D button
  - Choose the desired datatype from the list (e.g., int)
  - Provide the property name (e.g., person-age)
  - 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

 `person_has_age` <http://www.w3.org/2001/XMLSchema#int>



# New Datatype Property

The screenshot shows the GATE software interface with the 'Classes & Instances' and 'Properties' tabs. The 'Classes and Instances' tree shows a hierarchy: Entity (parent), Location, Organization, Person (child), and A\_Person (child of Person). The 'Properties' tab shows details for the 'Person' class, including its URI, type, and super classes (Entity).

The 'New Datatype Property' dialog box is open, showing the following fields:

- Name Space: `http://gate.ac.uk/example#`
- Data Type: `http://www.w3.org/2001/XMLSchema#int`
- Property Name: `person_has_age`

The 'Domain' dialog box is also open, showing a list of domains for the property:

- Selected Domain: `http://gate.ac.uk/example#Entity`
- Available Domain: `http://gate.ac.uk/example#Person`



# Object Properties

---

- Object properties link individuals to individuals (or instances)
- Describe relationships e.g. people work for organisations
- Multiple range restrictions are interpreted as stating that the range of the property is the intersection of all ranges
- 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

 `person_works_for [Organization]`



# New Object Property

The screenshot displays the GATE ontology editor interface. The main window is divided into several sections:

- Classes & Instances:** A tree view showing the ontology structure. The 'Entity' class is expanded, showing subclasses: 'Location', 'Organization' (with 'A\_Company' as a subclass), and 'Person' (with 'A\_Person' as a subclass).
- Properties:** A panel showing the configuration for the 'person\_works\_for' property. It includes:
  - Resource Information:** A table with columns for property name, URI, and type.

Person	Person	Person
URI	http://gate.ac.uk/example#Person	
TYPE	Ontology Class	
  - Direct Super Classes:** A list containing 'Entity'.
  - All Super Classes:** A list containing 'Entity'.

Overlaid on the main window are two dialog boxes:

- New Object Property:** A dialog for creating a new property. It has two input fields: 'Name Space' (containing 'http://gate.ac.uk/example#') and 'Property Name' (containing 'person\_works\_for'). There are 'Domain' and 'Range' buttons, and 'OK' and 'Cancel' buttons at the bottom.
- Domain:** A dialog for selecting the domain class. It features a dropdown menu with 'http://gate.ac.uk/example#Entity' selected. Below the dropdown are 'Add' and 'Remove' buttons. A list box below contains 'http://gate.ac.uk/example#Person', which is highlighted. 'OK' and 'Cancel' buttons are at the bottom.



# Object Properties in OAT

---

- To define a new object property
  - Click on the O button
  - Provide a property name
  - The domain is the class (or classes) that have this property as a subject
  - The range is the class (or classes) that have this property as an object
- For instances/individuals, set the value of an object property by:
  - Right-clicking on the instance
  - Selecting Properties/the-property-name, and
  - From the drop down list of instances choosing the correct instance as a value



# Hands-on 2

- 
- Use the entity ontology from the previous exercise
  - Model that locations have latitude and longitude (as datatype properties) and people have age
  - Add object properties to model that organisations have staff (who are people) and have offices (which are locations)
  - Add instances for each of these and provide some values for your properties
  - Save your resulting ontology to disk
    - Right click on the ontology, choose Save As, and choose a directory and file name
    - The RdfXml format is fine. You can open it in a text editor to see what the XML looks like

# 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*
  - *Reuse: swoogle (search engine for ontologies)*
  - Classes vs instances: what is the right granularity
    - Do I need subclasses of organisations (e.g., companies, charities, etc) for my application
  - Domains and ranges:
    - Choose them generic enough, but avoid the very top ones like Thing, as they will apply to every class
    - Do not provide both a class and its sub-class(es) as a domain/range, just the class will do
    - Avoid using unions of classes for domains/ranges

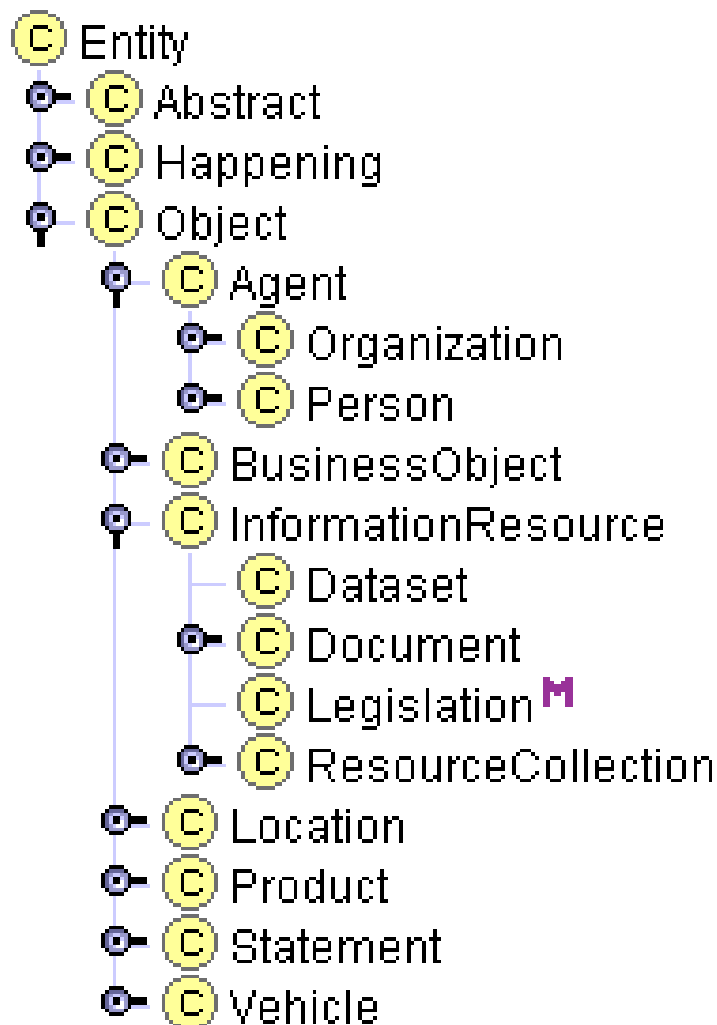




# PROTON Ontology

- a light-weight upper-level ontology
- 250 NE classes
- 100 relations and attributes
- covers mostly **NE classes**, and ignores general concepts

[proton.semanticweb.org](http://proton.semanticweb.org)





# Why ontologies in GATE?

---

- **Semantic annotation:** rather than just annotating the word “Cambridge” as a location, link it to an ontology instance
  - Cambridge, UK rather than Cambridge, Massachusetts, etc.
- **Semantic search via reasoning**
  - Ontologies tell us that this particular Cambridge is part of the country called the United Kingdom, which is part of the continent Europe.
  - So we can infer that this document mentions a city in Europe.
- **Knowledge source**
  - If I am looking to annotate strikes in baseball match reports, the ontology will tell me that strikes involve a batter who is a person
  - Faced with the text: “BA cabin crew went on strike” and using the knowledge that BA is a company and not a person, the IE system can conclude that this is not a strike event it needs to annotate



# Some Terminology

---

- **Ontology learning** – automatically derive an ontology from texts (typically limited to classes, class hierarchy and a small set of relations such as part-of)
- **Semantic annotation** – annotate in the texts all mentions of instances relating to concepts in the ontology (without modifying the ontology)
- **Ontology population** – given an ontology, populate it with instances derived automatically from a text



# 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

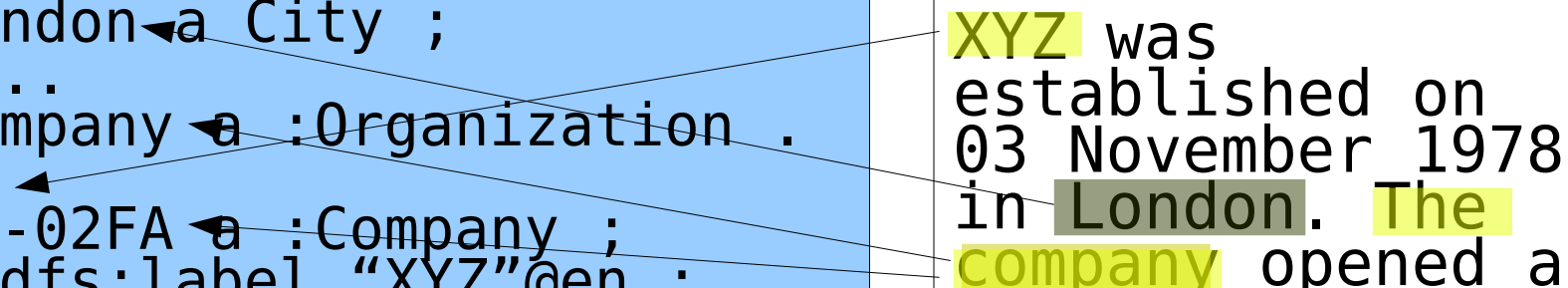
## Ontology

```

: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: Motivation

---



- Semantic annotation is the glue that ties ontologies into document spaces, via metadata
- Manual metadata production cost is too high
- Information extraction needs extending to target ontologies and scale to industrial document stores and the web

# Semantic Annotation vs IE



- 
- Traditional IE is based on a flat structure, e.g. recognising Person, Location, Organisation, Date, Time etc.
  - Semantic annotation attaches metadata to the documents, pointing to concepts and properties in an ontology
  - Information is typically exported as text annotated with links to the ontology



## Semantic Annotation: How?

---

- Manually: ontology based annotation – GATE OAT (Ontology Annotation Tool)
- Automatically
  - Gazetteer/rule/pattern based
  - Classifier (ML) 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: Options Tab

Ontology Tree(s) Options

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" and put "label" as the property
- To put all annotations into a set other than Default, change accordingly
- By default, OAT creates:
  - Annotations of type "Mention"
  - The "class" feature has the class URI
  - The "inst" feature – the instance URI if applicable

# OAT



The screenshot shows the GATE Developer 5.0 build 3244 interface. The main window displays a document with highlighted text. The ontology tree on the right shows various classes under the 'proton' ontology. A dialog box is open over the text 'Number', showing ontology and class information.

**Document 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.

"Compared with more of a price increase in the US, the UK has seen a fall in confidence in the economy and Alcatel's share price. In Germany, the government's commerce expansion showed consumer confidence at its lowest level for two years.

**Ontology Tree (proton):**

- JobPosition
- ContactInformation
- InformationResource
- Number
- Organization
  - PoliticalEntity
  - Team
  - Charity
  - EducationalOrganization
  - ReligiousOrganization
  - SportOrganization
  - GovernmentOrganization
  - StockExchange
  - Division
  - ResearchOrganization
  - CommercialOrganization
  - InternationalOrganization
- Document
- Location
- TimeInterval
- Event
- Agent

**Dialog Box:**

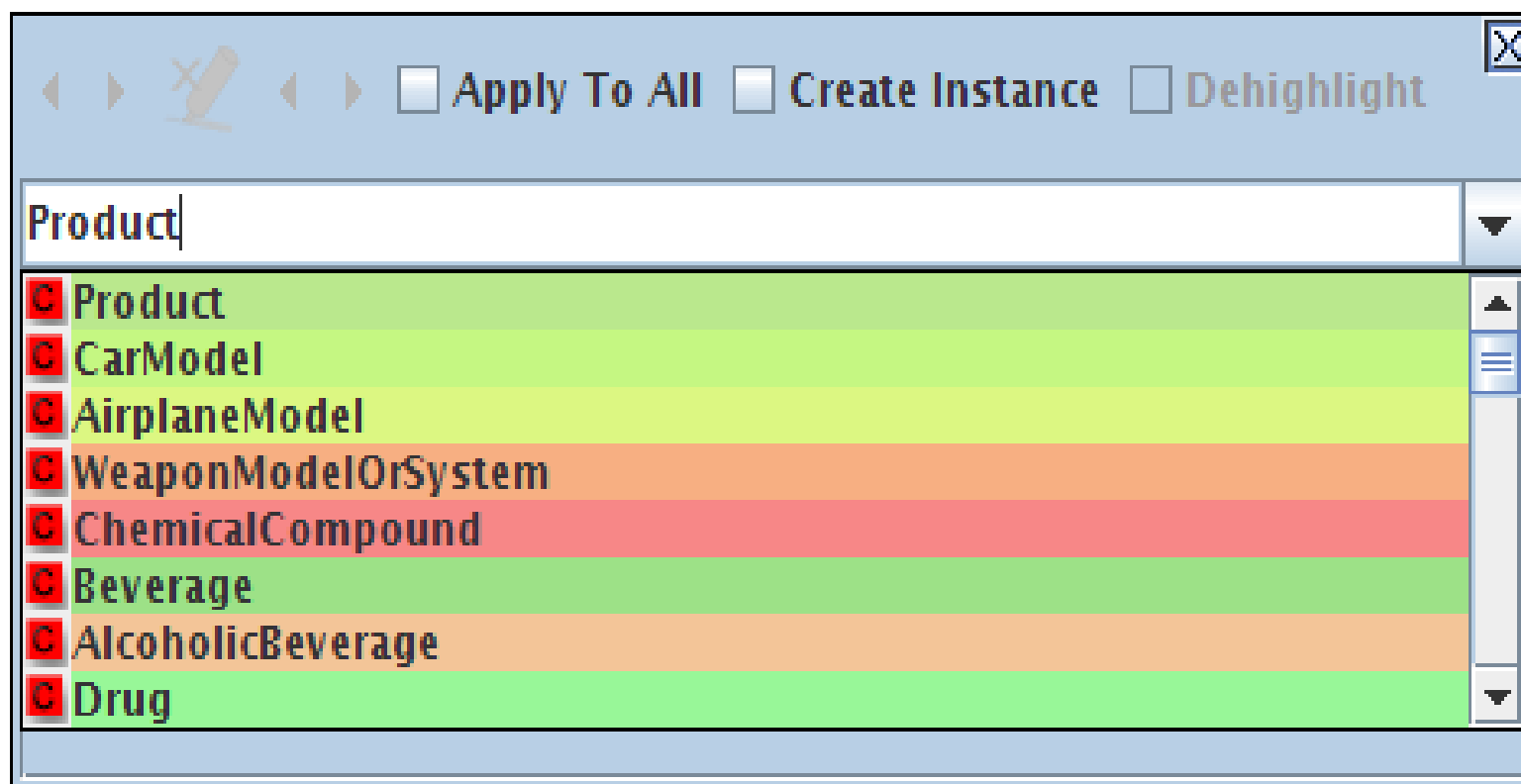
Number

ontology	http://proton.semanticweb.org/2005/04/proton	X
class	Number	X
label	[]	X
seeAlso	[]	X



# OAT: The Editor Pop-up

---





# Hands-on 3

- 
- (Load Ontology\_Tools plugin)
  - Use the previously created ontology or load it from disk
  - Load the sample document **voting-example.xml** (from hands-on)
  - Select the OAT button from the doc viewer
  - From the Options tab, choose Key as the target set and set labels to be created for new instances
  - Create annotations (e.g., UK is a location)
    - **Tip:** Use “apply to all” to annotate all mentions of UK in the text in one go
    - **Tip:** Check boxes and options need to be selected before choosing the target class
  - Experiment with automatically created instances in the ontology as you annotate (e.g., David Cameron). Switch to the ontology viewer to see the new instance
  - Examine the annotations created in Key and their features
  - Save back to disk



# OAT: More details

---

- The options to filter out some classes or only show some are useful when working with big ontologies
- Limitations:
  - Cannot annotate property values
  - Demo on Friday morning of the new editor forthcoming in GATE 6.0



# Semantic Annotation: Automatic

---

- Create language resources from an existing ontology:
  - Retrieve or generate possible mentions and create a gazetteer
- Semantic Annotation steps:
  - Pre-process document (e.g., tokeniser, morphology)
  - Annotate document with gazetteer
  - Disambiguation, post-processing



# Onto Root Gazetteer

---

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





# Onto Root Gazetteer

---

- Lives in the Gazetteer\_Ontology\_Based plugin
- Generate candidate list from ontology
- Runs the Tokeniser, POS tagger, Morphological Analyser (M.A.) to create lemmata from the labels and the URIs of all classes and instances and then creates lists to match against the text
  - Gordon\_Brown → Gordon Brown
- Created originally to match ontologies whose instances are terms, not just proper names, hence the need for such complex pre-processing of the labels and URIs, prior to using them as lists for matching



# Init-time OntoRoot params

Parameters for the new Onto Root Gazetteer

Name:

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

OK Help Cancel

Ontology LR

POS Tagger PR

Tokeniser PR



# Running OntoRoot

---

- 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, for generality to build 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
- It then matches the ontology URIs and labels against Token.root values in the text, which may capture more cases

# OntoRoot Application in GATE



Parameters for the new Flexible Gazetteer

Name:

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

OK Cancel Help

- Create a Flexible Gazetteer with an OntoRoot inside it

Selected Processing resources

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

- Build a GATE application with the PRs shown on the left



# Output Example

standing for election across the country.

David Cameron was the first of the main UK party leaders community hall in Witney, Oxfordshire, shortly after 1030 B

Lookup

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

Lookup

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

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
Lookup	880	886	9707	{ 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 4

- 
- Load plugin Gazetteer\_Ontology\_Based for OntoRoot
  - Plugin Tools for GATE Morphological Analyser and the Flexible Gazetteer
  - Create an OWLIM Ontology and load **test-ontology-instances.owl** from the hands-on directory
  - 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 4 (Continued)

---

- **Create pipeline**
  - \_ Add Tokeniser
  - \_ Create and add Sentence splitter
  - \_ Add POS Tagger
  - \_ Add Morphological Analyser
  - \_ Add Flexible Gazetteer
  - \_ Make all their input and output sets to be called **Test**
- Create a document from **voting-example.xml** (in hands-on dir)
- Add it to a corpus and provide this corpus to the pipeline
- Run
- Inspect the resulting Lookup annotations in the **Test** annotation set
- Save your application for later use and keep it open



# 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



# Conventions in GATE for SA



- 
- 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)
  - Example (with the name spaces omitted):
    - Mention {class=Leader, inst=Gordon\_Brown}



# Ontology Aware JAPE

---

- JAPE transducers have a run-time parameter which is an ontology
- By default it is left blank, so not used during LHS matching
- When provided and the **class** feature is used on the LHS, then when matching its value the ontology is checked for subsumption
  - e.g., `Lookup.class = Person` will match a `Lookup` annotation with **class** feature, whose value is a sub-class of `Person` (as well as `Person` itself)



# Ontology-aware JAPE

---

```
Phase: OntoMatching
Input: Lookup
Options: control = appelt
```

```
Rule: PersonLookup
```

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

*Matches any name of a class that is a subclass of Person, as well as Person*

Best is to update GATE from svn, after 5.2.1



# An Example

David Cameron was the first...



Lookup

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

This Lookup annotation of class Leader will match the LHS of our rule, because Person is a super-class of Leader



# BUT...

- 
- OntoRoot gazetteer puts the class URIs in a feature called **classURI** and the instance URI in a **URI** feature
  - Therefore we need a JAPE grammar to go through each Lookup annotation and copy the value of its **classURI** feature to a **class** feature
    - See previous slide for an example where this has been done



# This JAPE grammar...

```
Phase: LookupRename
Input: Lookup
Options: control = appelt

Rule: RenameLookup
(
  {Lookup.type == instance}
):match
-->
:match{
  For (Annotation lookup : matchAnnots) {
    FeatureMap theFeatures = lookup.getFeatures();
    theFeatures.put(
      "class", theFeatures.get("classURI"));
    theFeatures.put("inst", theFeatures.get("URI"));
  }
}
```

...finds all **Lookup** annotations produced by **OntoRoot** which are of **type=instance**, takes the value of their **classURI** feature and copies it to the **class** features. And similarly for the instance **URI**, which is copied to **inst**



# Hands-on 5

---

- Modify the pipeline from hands-on 4
  - Add a Document Reset PR at the start and set its setsToRemove run-time param to [Test]
  - Create a JAPE transducer for **rename-lookup-features.jape** and add as the last PR (input and output sets should be set to Test)
- Run the modified pipeline to see how some of the Lookup annotations in Test now have class features
- Create a JAPE transducer for **person-onto-matching.jape** and put it last in the pipeline and make sure inputs and outputs are in the Test annotation set. Give the ontology as the run-time param
- Run the modified pipeline to see how it creates new **Mention** annotations
- Save the application (under a new name to keep the older copy) and close it
- Continue to the next exercise



# Hands-on 6

---

- Modify the ontology:
  - Add a subclass Country of the Location class
  - Add an instance “UK” of Country
  - **Tip:** If you have already UK as an instance of Location, you need to delete it first and then re-add it.
- In a text editor open **person-onto-matching.jape** and add a similar rule that matches any location and creates a Mention annotation
- Within GATE reload your saved application and re-run it to make sure it now annotates UK as a Mention too



# Beyond Gazetteer-based Annotation



- 
- 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 list-based gazetteers with Lookups from ontologies and other clues, in order to detect Mentions
  - Co-reference: Mr. Brown = Gordon Brown = he
  - Disambiguation: if several instances with label “John Smith” pick the correct one
    - Context from the ontology can be matched against the text



# Performance Evaluation

---

- Mention annotations can be evaluated against a gold standard by matching the class attribute (and inst, if available)
- AnnotDiff and Corpus QA require that the values are identical, otherwise it is a spurious annotation
- However, some mistakes can be “more wrong” than others
  - Nick Clegg → Person (not Leader) – still logically correct
  - Nick Clegg → Location – wrong
- Therefore the class hierarchy needs to be taken into account when calculating precision and recall

# Use OAT to create gold standard



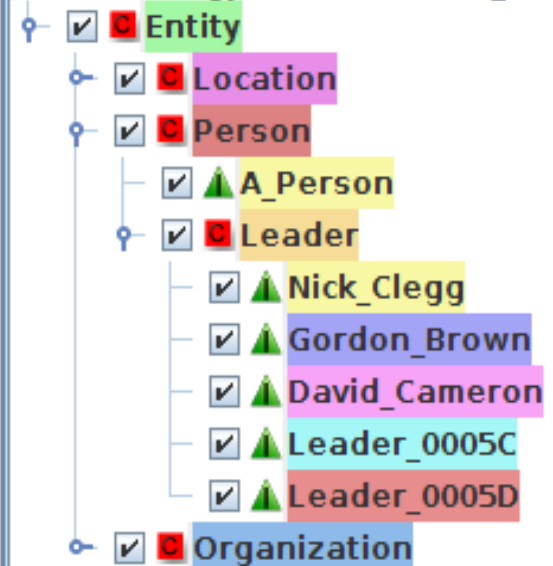
David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

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\_022:



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.

Here we have added two new instances (Salmond and Jones) as well as annotating the text



# Traditional Precision/Recall

Annotation Diff Tool

Key doc: voting-example.xml... Key set: Key Type: Mention Weight: 1.0

Resp. doc: voting-example.xml... Resp. set: result Features:  all  some  none

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}

I have edited manually the system results and made Clegg of class Person

Correct: 2 Recall Precision F-measure

Partially correct: 0 Strict: 0.40 0.67 0.50

Missing: 3 Lenient: 0.40 0.67 0.50

False positives: 1 Average: 0.40 0.67 0.50

Statistics Adjudication

1 documents loaded

Show document

Export to HTML



# Balanced Distance Metric

---

- BDM measures the closeness of two concepts in an ontology or taxonomy
- It is a real number between 0 and 1
- The closer the two concepts are in an ontology, the greater their BDM score is
- It is dependent on the length of the shortest path connecting the two concepts and also the depth of the two concepts in the ontology
- It is also normalized according to the size of the ontology and its density



# 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**
  - If you run BDM on your own ontology, please do not over-write this file, we'll need it later

```
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
```



# Precision, Recall, F-score

---

$$Precision = \frac{Correct}{Correct + S}$$

$$Recall = \frac{Correct}{Correct + A}$$

$$F = 2 \cdot \left( \frac{precision \cdot recall}{precision + recall} \right)$$

# Ontology-sensitive F-scores



- 
- The IAA plugin computes precision, recall, and F-score over a corpus
  - Takes the BDM output from the BDM PR
  - Uses it to compute ontology-sensitive Precision, Recall, and F-score
  - Details on BDM and how it affects these scores:
    - D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In WWW 2006 Workshop on Evaluation of Ontologies for the Web (EON), Edinburgh, Scotland, 2006.





# Running IAA with BDM

---

- Load the Inter\_Annotator\_Agreement Plugin
- Load the IAA Computation PR
- Create a corpus pipeline and put it in
- The parameters are a bit idiosyncratic
  - AnnSetsforlaa: Key;Test
  - AnnTypesAndFeats: Mention->class
  - BdmScoreFile: bdm-output.txt
  - FMeasure and verbosity=1
- Run it on the corpus
- Prints the results in the Messages tab



# Our example text again

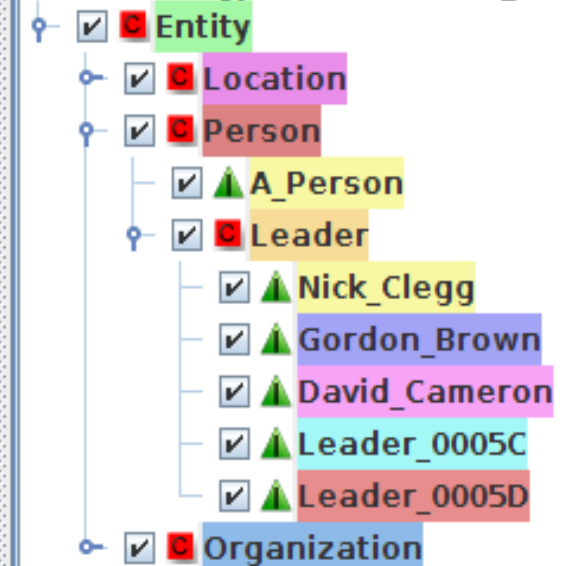
David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

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 and Jones are missing



# Sample Output

---

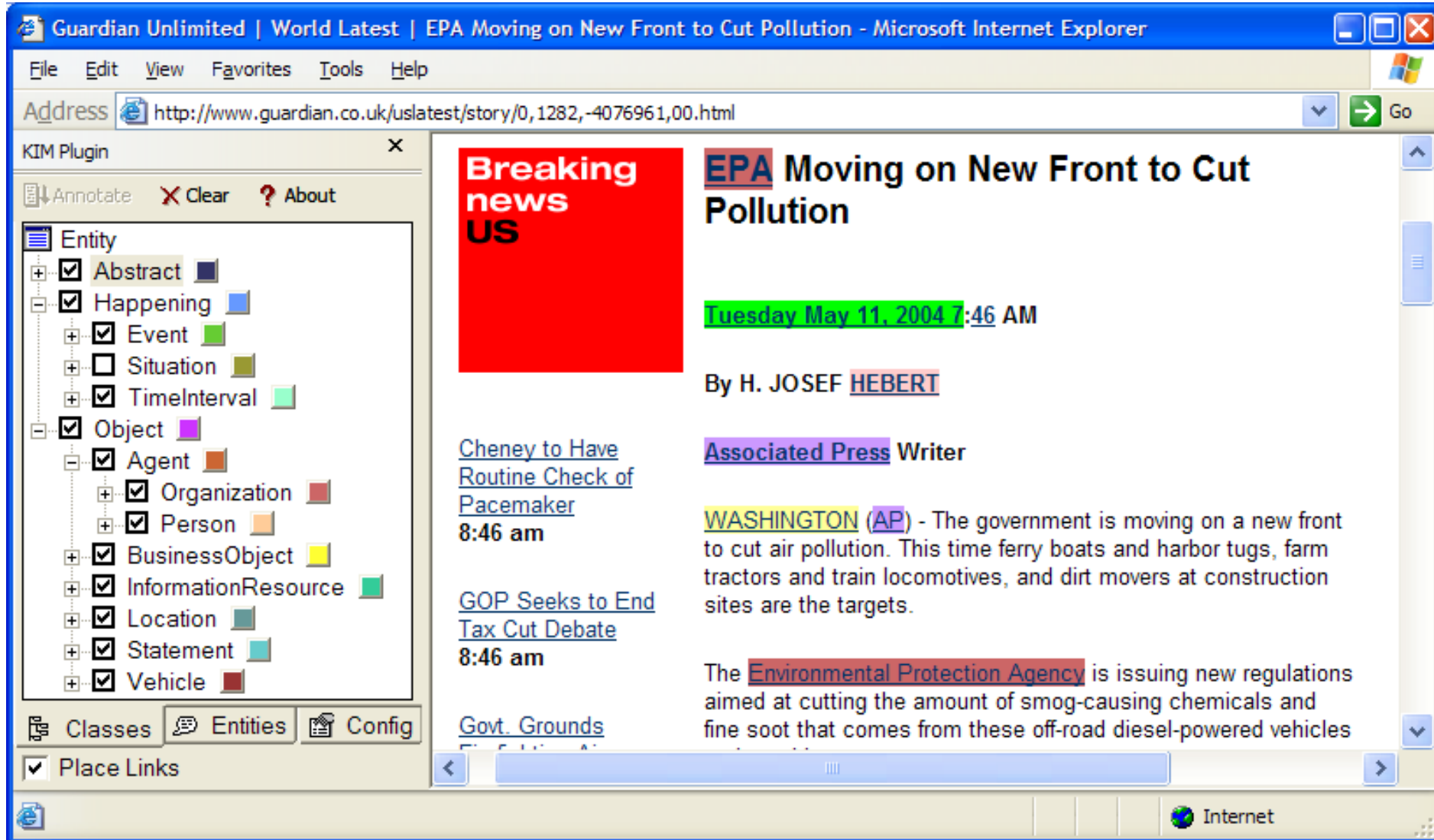
- This sample output is from the voting-example document, with the small 5 class ontology and the BDM output supplied in `bdm-output.txt`
- The traditional scores again:
  - (correct, partial, spurious, missing)= (2.0, 0.0, 1.0, 3.0)
  - (precision, recall, F1)= (0.6666667, 0.4, 0.5)
- BDM-sensitive scores:
  - (correct, partial, spurious, missing)= (2.4186046, 0.0, 0.0, 2.0)
  - (precision, recall, F1)= (0.806201533, 0.5473684, 0.707483)
  - **NB: There is a bug above in the calculation of spurious and missing and the PR is being fixed**

# Sem. Annotation Example



- 
- The KIM system
    - Performs semantic annotation
    - Provides tools for semantic indexing and search
    - Scales up to millions of documents
    - Visualisations over time
    - <http://ln.ontotext.com/kim/>

# Simple Usage: Highlight, Hyperlink, and ...



Guardian Unlimited | World Latest | EPA Moving on New Front to Cut Pollution - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.guardian.co.uk/uslatest/story/0,1282,-4076961,00.html> Go

KIM Plugin

Annotate Clear About

Entity

- Abstract
- Happening
  - Event
  - Situation
  - TimeInterval
- Object
  - Agent
    - Organization
    - Person
  - BusinessObject
  - InformationResource
  - Location
  - Statement
  - Vehicle

Classes Entities Config

Place Links

**Breaking news US**

## EPA Moving on New Front to Cut Pollution

Tuesday May 11, 2004 7:46 AM

By H. JOSEF [HEBERT](#)

[Associated Press](#) Writer

[WASHINGTON \(AP\)](#) - The government is moving on a new front to cut air pollution. This time ferry boats and harbor tugs, farm tractors and train locomotives, and dirt movers at construction sites are the targets.

The [Environmental Protection Agency](#) is issuing new regulations aimed at cutting the amount of smog-causing chemicals and fine soot that comes from these off-road diesel-powered vehicles

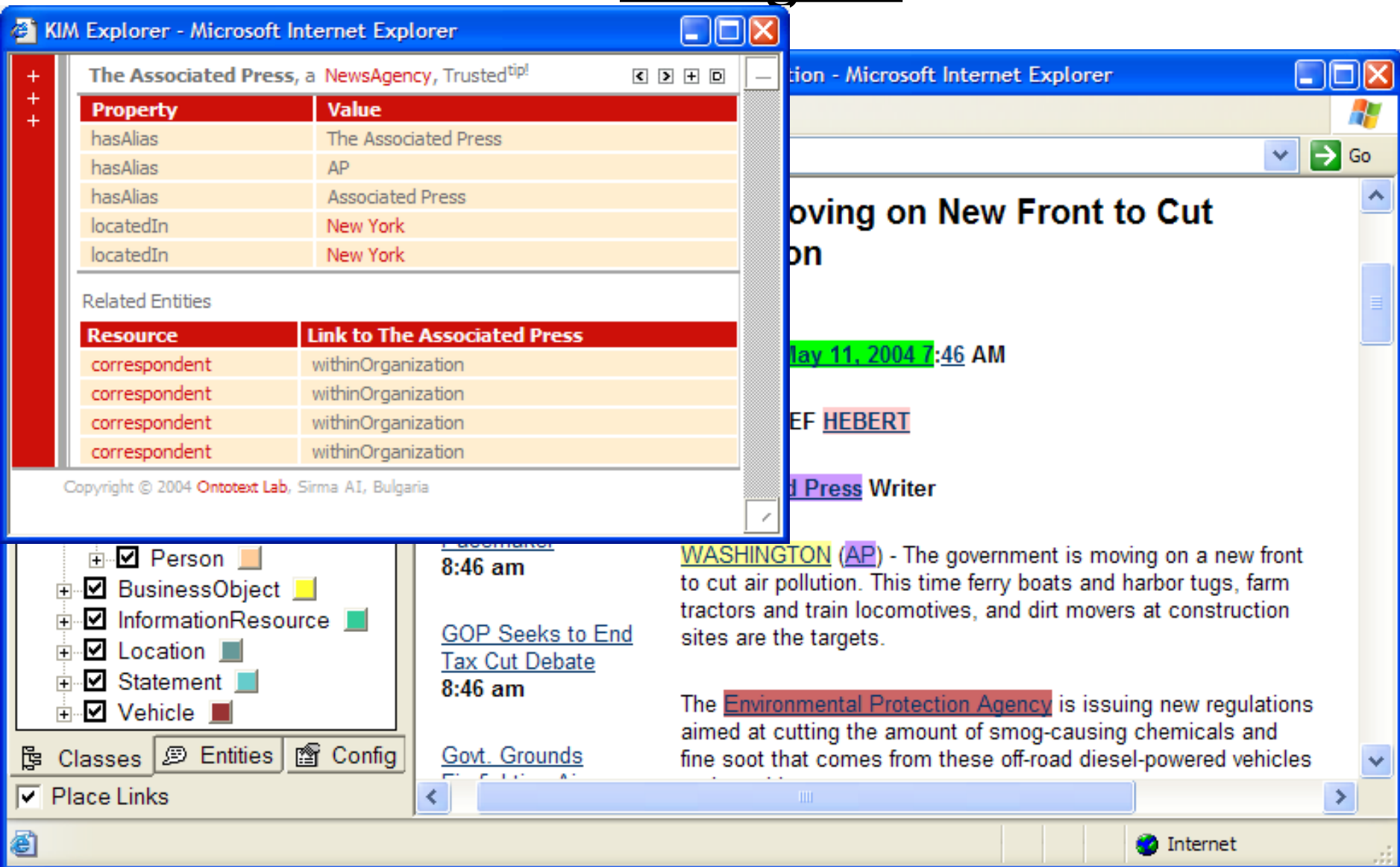
[Cheney to Have Routine Check of Pacemaker](#)  
8:46 am

[GOP Seeks to End Tax Cut Debate](#)  
8:46 am

[Govt. Grounds](#)

Internet

# Simple Usage: ... Explore and Navigate



The screenshot displays two overlapping windows. The foreground window is 'KIM Explorer - Microsoft Internet Explorer', which shows a semantic structure for 'The Associated Press, a NewsAgency, TrustedTip!'. It features a table of properties and values, and a section for related entities.

Property	Value
hasAlias	The Associated Press
hasAlias	AP
hasAlias	Associated Press
locatedIn	New York
locatedIn	New York

Resource	Link to The Associated Press
correspondent	withinOrganization
correspondent	withinOrganization
correspondent	withinOrganization
correspondent	withinOrganization

Copyright © 2004 Ontotext Lab, Sirma AI, Bulgaria

The background window is 'Microsoft Internet Explorer' displaying a news article. The article title is partially visible: '...oving on New Front to Cut ... on'. The date is 'May 11, 2004 7:46 AM'. The author is 'HEBERT'. The article text includes: 'WASHINGTON (AP) - The government is moving on a new front to cut air pollution. This time ferry boats and harbor tugs, farm tractors and train locomotives, and dirt movers at construction sites are the targets.' and 'The Environmental Protection Agency is issuing new regulations aimed at cutting the amount of smog-causing chemicals and fine soot that comes from these off-road diesel-powered vehicles'.

At the bottom left, there is a 'Classes' panel with a tree view showing checked items: Person, BusinessObject, InformationResource, Location, Statement, and Vehicle. Below it are buttons for 'Classes', 'Entities', and 'Config', and a checkbox for 'Place Links'.

# Entity Pattern Search

KIM WEB UI - Microsoft Internet Explorer

File Edit View Favorites Tools Help




Address <http://ontotest.sirma.bg/KIM/screen/EntityPatternSearch.jsp> Go

**KIM**

- > Datastore
- > Entity Pattern Search
- > Predefined Patterns
- > Entity Lookup
- > Keyword Search
- > About KIM

---

Powered by:

## Pattern Search

- Lookup for patterns where**

X, is a  , which name

and X  Y

Y, is a  , which name

and   Z

Z, is a  , which name
- attribute restrictions:**

Z

X

Y
- Interested In:**
- Search for:**

# Pattern Search: Entity Results

KIM WEB UI - Microsoft Internet Explorer

File Edit View Favorites Tools Help




Address <http://ontotest.sirma.bg/KIM/screen/EntitiesResult.jsp> Go

**KIM**

- > Datastore
- > Entity Pattern Search
- > Predefined Patterns
- > Entity Lookup
- > Keyword Search
- > About KIM

---

Powered by:

## Entity Query Result

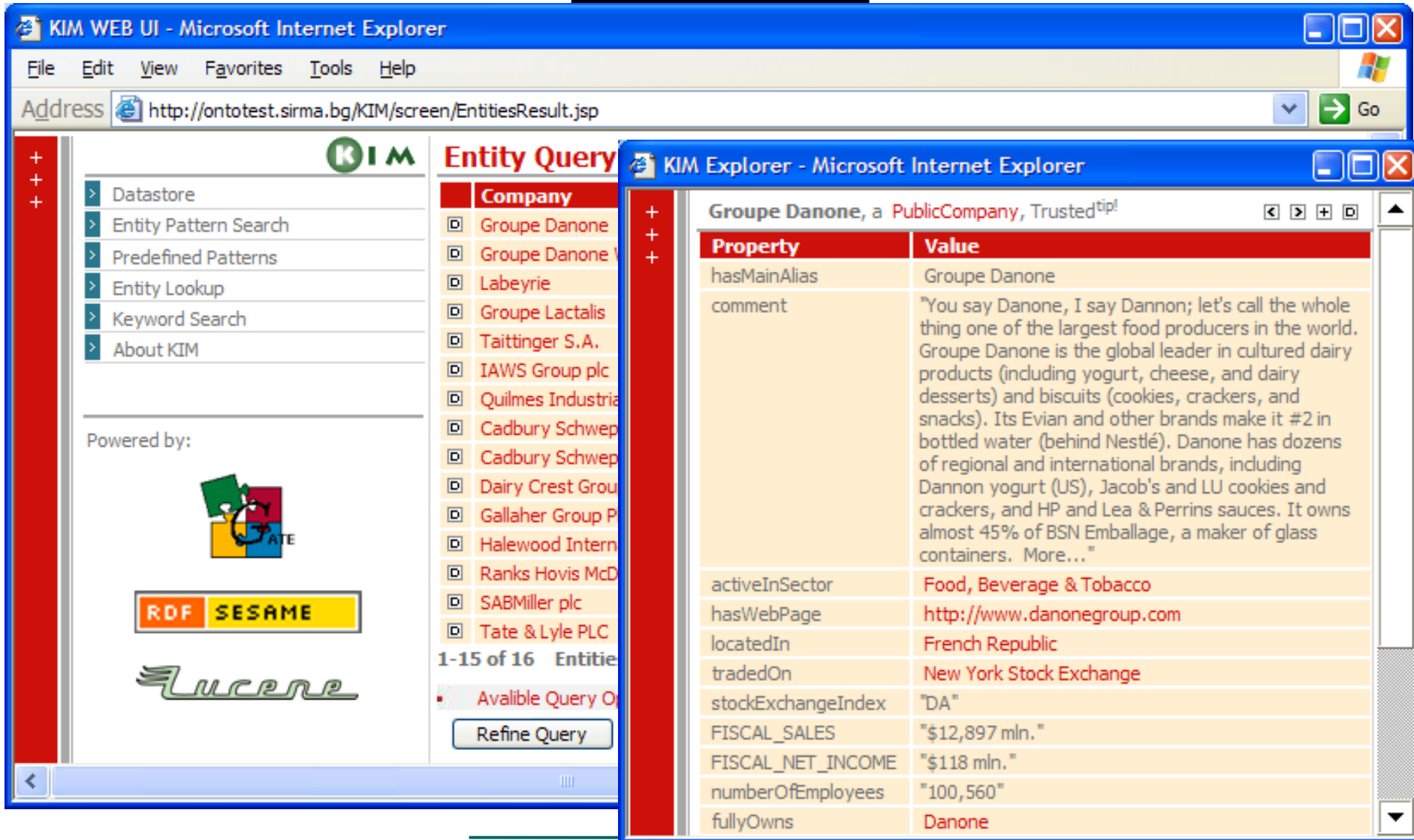
	Company	Type	Tr
<input type="checkbox"/>	Groupe Danone	PublicCompany	+
<input type="checkbox"/>	Groupe Danone World Water Division	Company	+
<input type="checkbox"/>	Labeyrie	Company	+
<input type="checkbox"/>	Groupe Lactalis	Company	+
<input type="checkbox"/>	Taittinger S.A.	Company	+
<input type="checkbox"/>	IAWS Group plc	PublicCompany	+
<input type="checkbox"/>	Quilmes Industrial S.A.	PublicCompany	+
<input type="checkbox"/>	Cadbury Schweppes Beverage Unit	Company	+
<input type="checkbox"/>	Cadbury Schweppes plc	PublicCompany	+
<input type="checkbox"/>	Dairy Crest Group plc	PublicCompany	+
<input type="checkbox"/>	Gallaher Group Plc	PublicCompany	+
<input type="checkbox"/>	Halewood International Limited	Company	+
<input type="checkbox"/>	Ranks Hovis McDougall Limited	Company	+
<input type="checkbox"/>	SABMiller plc	PublicCompany	+
<input type="checkbox"/>	Tate & Lyle PLC	PublicCompany	+

1-15 of 16 Entities per page:

Available Query Options



# Entity Pattern Search: KIM Explorer



The screenshot shows two overlapping browser windows. The background window is 'KIM WEB UI - Microsoft Internet Explorer' displaying the 'Entity Query' results page. The foreground window is 'KIM Explorer - Microsoft Internet Explorer' displaying a detailed view of the 'Groupe Danone' entity.

**KIM WEB UI - Entity Query Results:**

- Company
  - Groupe Danone
  - Groupe Danone
  - Labeyrie
  - Groupe Lactalis
  - Taittinger S.A.
  - IAWS Group plc
  - Quilmes Industria
  - Cadbury Schwepp
  - Cadbury Schwepp
  - Dairy Crest Group
  - Gallaher Group Pl
  - Halewood Intern
  - Ranks Hovis McD
  - SABMiller plc
  - Tate & Lyle PLC

1-15 of 16 Entities

Available Query Options

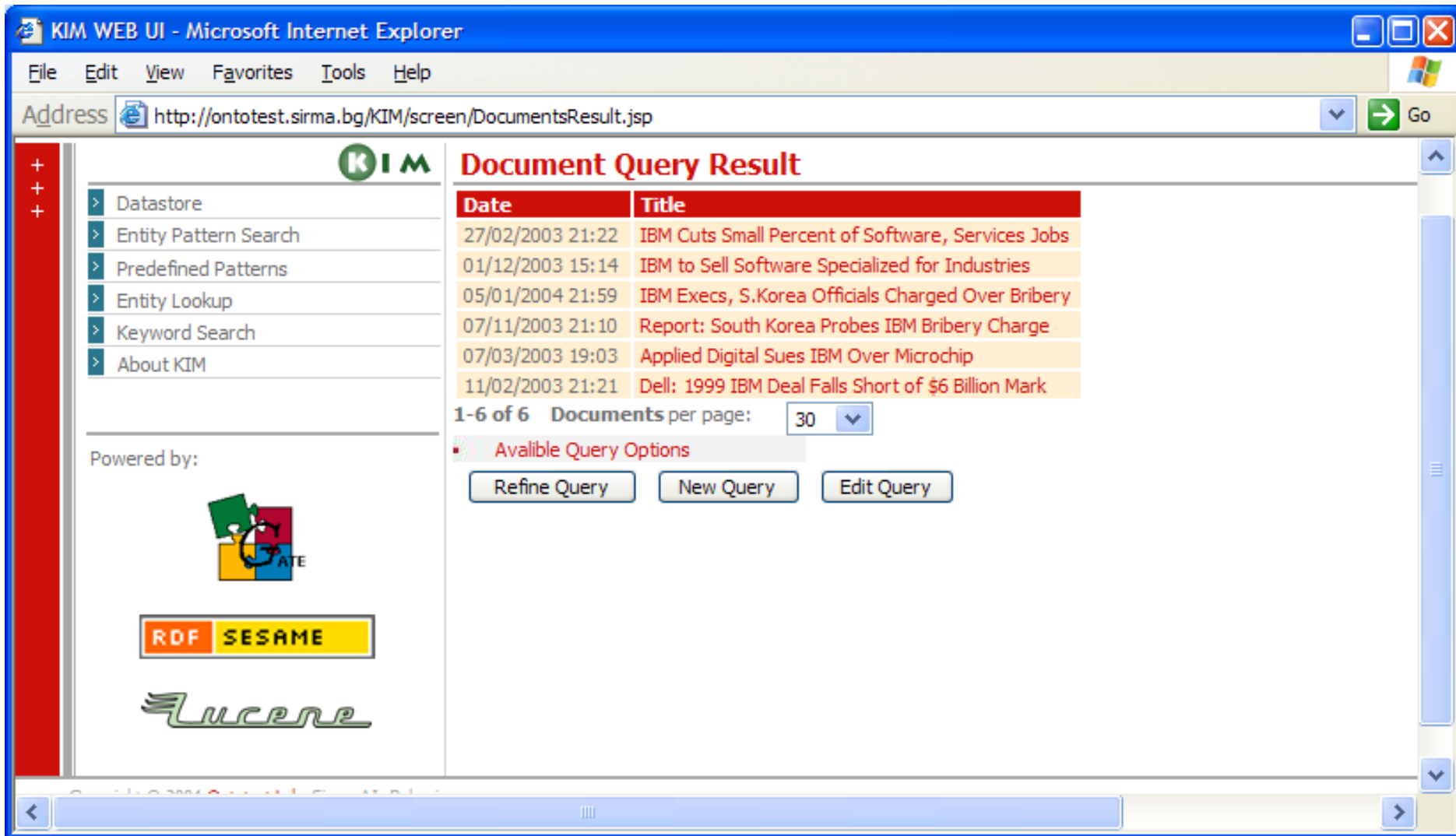
Refine Query

**KIM Explorer - Detailed View:**

Groupe Danone, a **PublicCompany**, Trusted<sup>tip!</sup>

Property	Value
hasMainAlias	Groupe Danone
comment	"You say Danone, I say Dannon; let's call the whole thing one of the largest food producers in the world. Groupe Danone is the global leader in cultured dairy products (including yogurt, cheese, and dairy desserts) and biscuits (cookies, crackers, and snacks). Its Evian and other brands make it #2 in bottled water (behind Nestlé). Danone has dozens of regional and international brands, including Dannon yogurt (US), Jacob's and LU cookies and crackers, and HP and Lea & Perrins sauces. It owns almost 45% of BSN Emballage, a maker of glass containers. More..."
activeInSector	Food, Beverage & Tobacco
hasWebPage	<a href="http://www.danonegroup.com">http://www.danonegroup.com</a>
locatedIn	French Republic
tradedOn	New York Stock Exchange
stockExchangeIndex	"DA"
FISCAL_SALES	"\$12,897 mln."
FISCAL_NET_INCOME	"\$118 mln."
numberOfEmployees	"100,560"
fullyOwns	Danone

# Pattern Search, Referring Documents



The screenshot shows a web browser window titled "KIM WEB UI - Microsoft Internet Explorer". The address bar contains the URL "http://ontotest.sirma.bg/KIM/screen/DocumentsResult.jsp". The main content area displays a "Document Query Result" table with the following data:

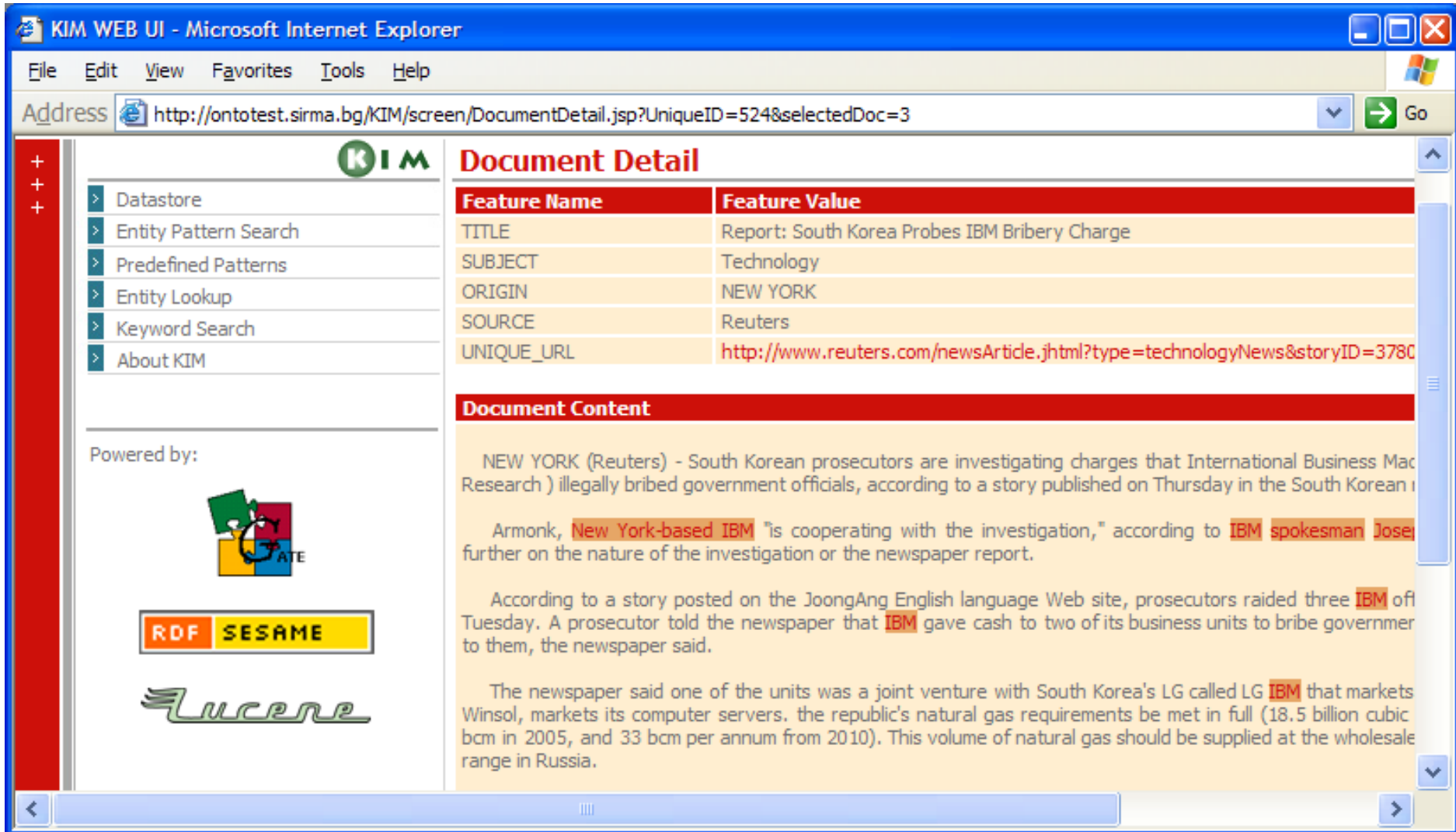
Date	Title
27/02/2003 21:22	IBM Cuts Small Percent of Software, Services Jobs
01/12/2003 15:14	IBM to Sell Software Specialized for Industries
05/01/2004 21:59	IBM Execs, S.Korea Officials Charged Over Bribery
07/11/2003 21:10	Report: South Korea Probes IBM Bribery Charge
07/03/2003 19:03	Applied Digital Sues IBM Over Microchip
11/02/2003 21:21	Dell: 1999 IBM Deal Falls Short of \$6 Billion Mark

Below the table, it indicates "1-6 of 6 Documents per page: 30" and provides "Available Query Options" with buttons for "Refine Query", "New Query", and "Edit Query".

The left sidebar contains navigation links: "Datstore", "Entity Pattern Search", "Predefined Patterns", "Entity Lookup", "Keyword Search", and "About KIM".



At the bottom left, logos for "Powered by:" include GATE, RDF, SESAME, and Lucene.

# Document Details



KIM WEB UI - Microsoft Internet Explorer




File Edit View Favorites Tools Help

Address  http://ontotest.sirma.bg/KIM/screen/DocumentDetail.jsp?UniqueID=524&selectedDoc=3 

**KIM**

- > Dastore
- > Entity Pattern Search
- > Predefined Patterns
- > Entity Lookup
- > Keyword Search
- > About KIM

Powered by:

## Document Detail

Feature Name	Feature Value
TITLE	Report: South Korea Probes IBM Bribery Charge
SUBJECT	Technology
ORIGIN	NEW YORK
SOURCE	Reuters
UNIQUE_URL	<a href="http://www.reuters.com/newsArticle.jhtml?type=technologyNews&amp;storyID=3780">http://www.reuters.com/newsArticle.jhtml?type=technologyNews&amp;storyID=3780</a>

## Document Content

NEW YORK (Reuters) - South Korean prosecutors are investigating charges that International Business Machines Corp. (IBM) illegally bribed government officials, according to a story published on Thursday in the South Korean newspaper JoongAng Ilbo.

Armonk, New York-based IBM "is cooperating with the investigation," according to IBM spokesman Jose L. Belfrage. He did not provide further on the nature of the investigation or the newspaper report.

According to a story posted on the JoongAng English language Web site, prosecutors raided three IBM offices in South Korea on Tuesday. A prosecutor told the newspaper that IBM gave cash to two of its business units to bribe government officials, the newspaper said.

The newspaper said one of the units was a joint venture with South Korea's LG called LG IBM that markets Winsol, markets its computer servers. The republic's natural gas requirements be met in full (18.5 billion cubic meters in 2005, and 33 bcm per annum from 2010). This volume of natural gas should be supplied at the wholesale range in Russia.



# 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

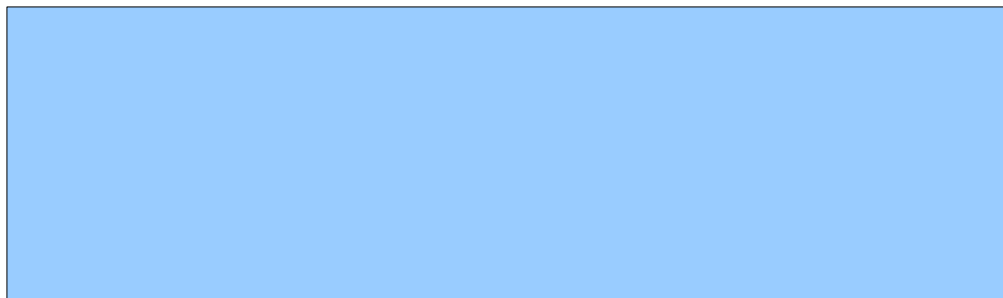


# Ontology population

---

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

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





# Ontology population

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

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



# Ontology population

---

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

```
:XYZ-001 a :Company ;  
:established-in :London .
```

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



# Ontology Population

---

- Populate Ontology with Instances:
  - Of classes
  - Of properties connecting class instances with other class instances or values (literals)
- Examples and further information in Module 6, since it requires Java programming and good knowledge of the ontology API





# 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, ...



# 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.

# 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 )

# 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" }
```



# 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.