WP 2: Learning Web-service Domain Ontologies

Miha Grčar

Jožef Stefan Institute

http://www.tao-project.eu

Funded by: European Commission – 6th Framework
Project Reference: IST-2004-026460

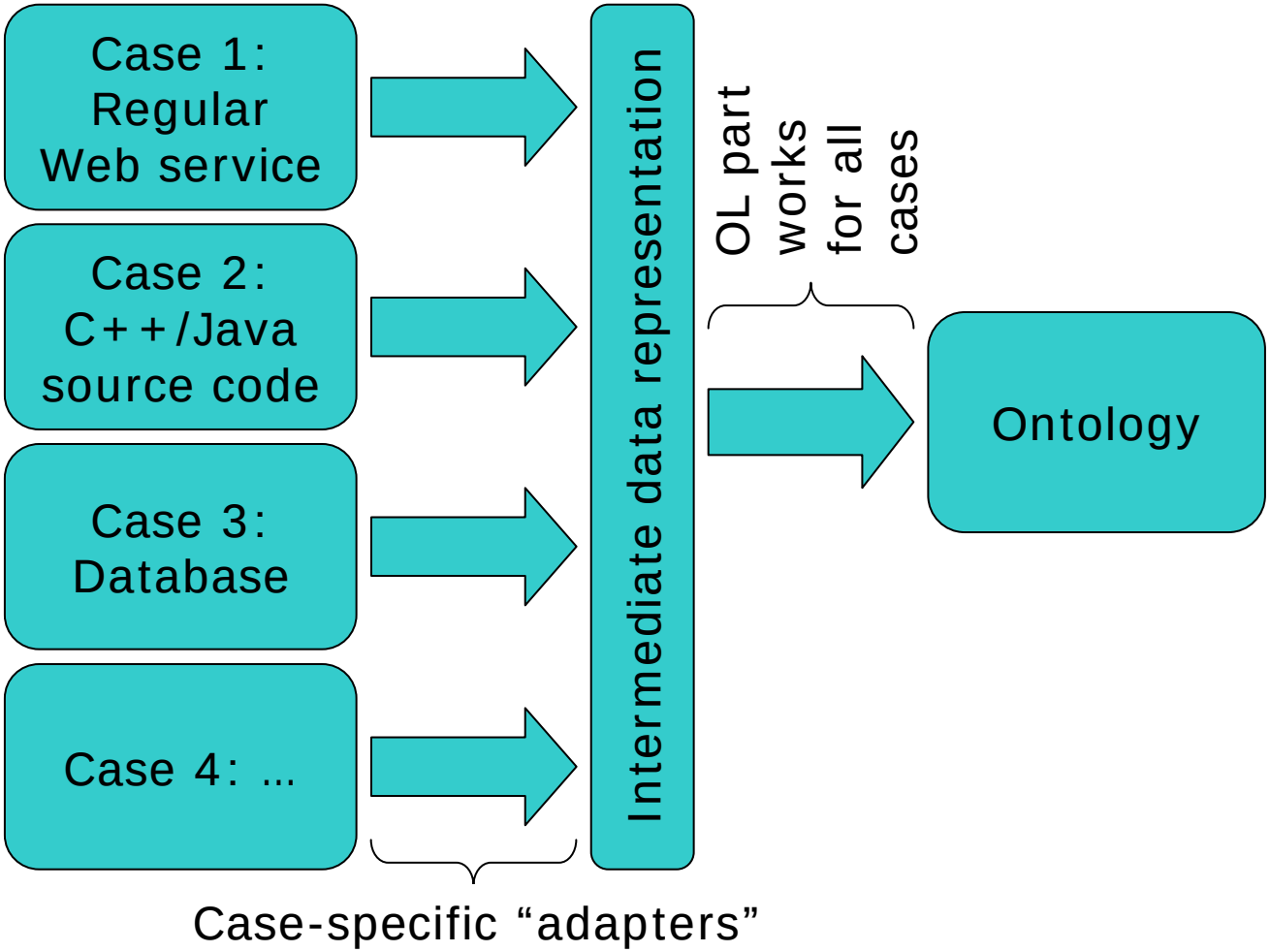
Outline of the Presentation

- The goal of WP 2
- Introduction to application mining
- Creating a document network
- Transforming a document network into feature vectors
- LATINO: Link-analysis and text-mining toolbox
- OntoGen: a system for semi-automatic data-driven ontology construction
- WP 2 and the Dassault case study
- Conclusions and future work

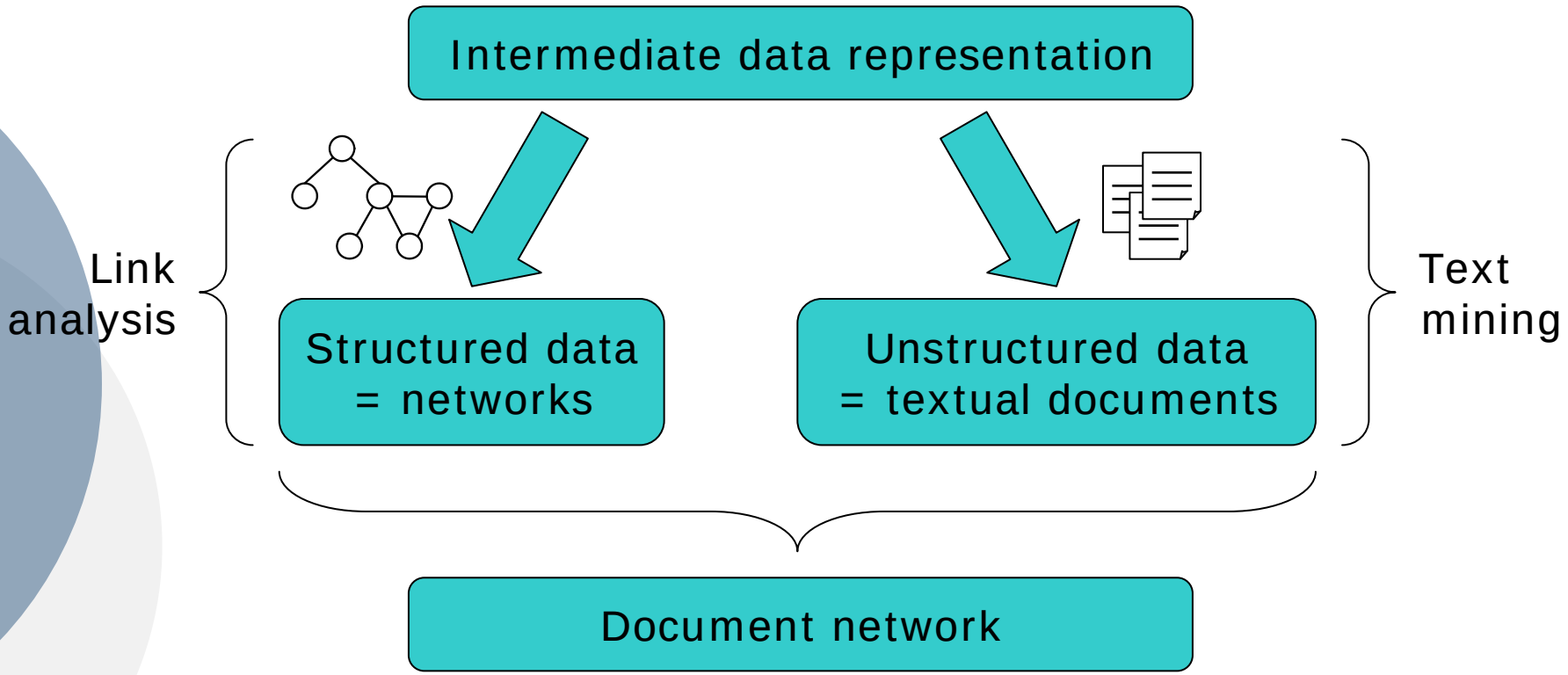
Learning Web-service Ontologies

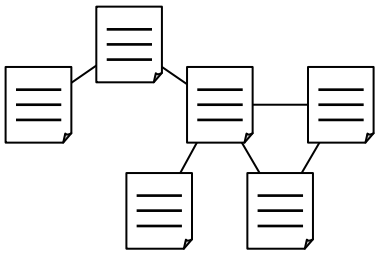
- The goal is to facilitate the acquisition of domain ontologies from legacy applications by:
  1. Identifying data sources that contain knowledge to be transitioned into an ontology
  2. Employing data mining techniques to aid the domain expert in building the ontology

Application Mining



Application Mining





= A set of interlinked documents; each link has a type and a weight

GATE Case Study

- Software library for natural language processing (NLP)
- ~600 Java classes
  - Language resources = data
  - Processing resources = algorithms
  - Graphical user interfaces = GUI
- Developed at University of Sheffield
- Freely available at http://gate.ac.uk/download/

Data Sources

- Structured
  - Code samples
  - Web service usage logs
  - Source code
  - Reference manual (function declarations)
  - WDSL ...
- Unstructured
  - Web pages
  - User's manual
  - Tutorials, lectures, forums, newsgroups, etc.
  - Reference manual (textual descriptions)
  - Source code comments ...

A Typical Java Class

Class comment

Comment references

```
/** The format of Documents. Su umentFormat know about
  * particular MIME types and how to unpack the information in any
  * markup or formatting they contain into GATE annotations. Each MIME
  * type has its own subclass of DocumentFormat, e.g. XmlDocumentFormat,
  * RtfDocumentFormat, MpegDocumentFormat. These classes register themselves
  * with a static index residing here when they are constructed. Static
  * getDocumentFormat methods can then be used to get the appropriate
  * format class for a particular document.
  */
public abstract class DocumentFormat
extends AbstractLanguageResource implements LanguageResource{
```

Class name

Super-class (base class)

Implemented interface

A field

```
/** The MIME type of this format. */
 private MimeType mimeType = null;
```

Field comment

Field type

Field name

A method

Method comment

```
/**
  * Find a DocumentFormat implementation that deals with a particular
  * MIME type, given that type.
  * @param  aGateDocument this document will receive as a feature
  *                       the associated Mime Type. The name of the feature is
  *                       MimeType and the format type/subtype
  * @param  mimeType the mime type that is given as input
  */
 static public DocumentFormat getDocumentFormat(gate.Document aGateDocument,
                                                MimeType mimeType){
```

Comment reference

Return type

Method name

```
  } // getDocumentFormat(aGateDocument, MimeType)

} // class DocumentFormat
```

Creating a Document Network

DocumentFormat.class

```
/** The format of Documents. Subclasses of DocumentFormat know about
 * particular MIME types and how to unpack the information in any
 * markup or formatting they contain into GATE annotations. Each MIME
 * type has its own subclass of DocumentFormat, e.g. XmlDocumentFormat,
 * RtfDocumentFormat, MpegDocumentFormat. These classes register themselves
 * with a static index residing here when they are constructed. Static
 * getDocumentFormat methods can then be used to get the appropriate
 * format class for a particular document.
 */
public abstract class DocumentFormat
extends AbstractLanguageResource implements LanguageResource{
```

```
/** The MIME type of this format. */
 private MimeType mimeType = null;
```

```
/**
   * Find a DocumentFormat implementation that deals with a particular
   * MIME type, given that type.
   * @param  aGateDocument this document will receive as a feature
   *                       the associated Mime Type. The name of the feature is
   *                       MimeType and its value is in the format type/subtype
   * @param  mimeType the mime type that is given as input
   */
 static public DocumentFormat getDocumentFormat(gate.Document aGateDocument,
                                                MimeType mimeType){
```

```
 } // getDocumentFormat(aGateDocument, MimeType)

} // class DocumentFormat
```

DocumentFormat

Creating a Document Network

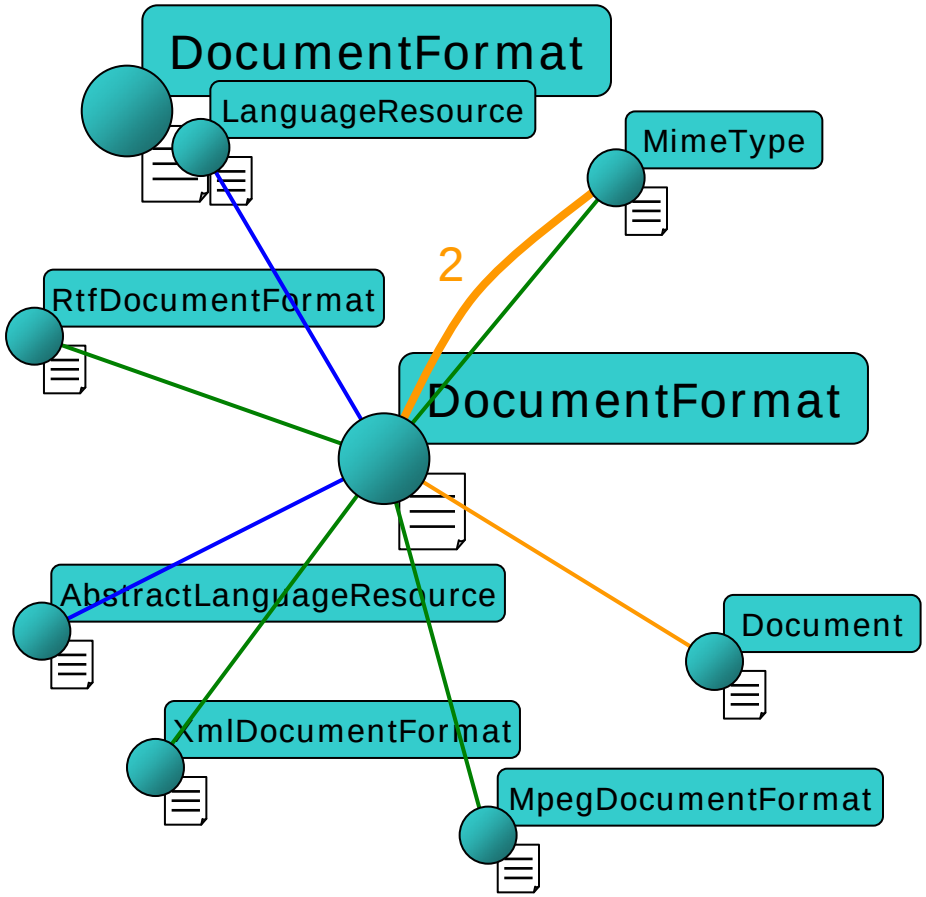
DocumentFormat.class

```
/** The format of Documents. Subclasses of DocumentFormat know about
 * particular MIME types and how to unpack the information in any
 * markup or formatting they contain into GATE annotations. Each MIME
 * type has its own subclass of DocumentFormat, e.g. XmlDocumentFormat,
 * RtfDocumentFormat, MpegDocumentFormat. These classes register themselves
 * with a static index residing here when they are constructed. Static
 * getDocumentFormat methods can then be used to get the appropriate
 * format class for a particular document.
 */
public abstract class DocumentFormat
extends AbstractLanguageResource implements LanguageResource{
```

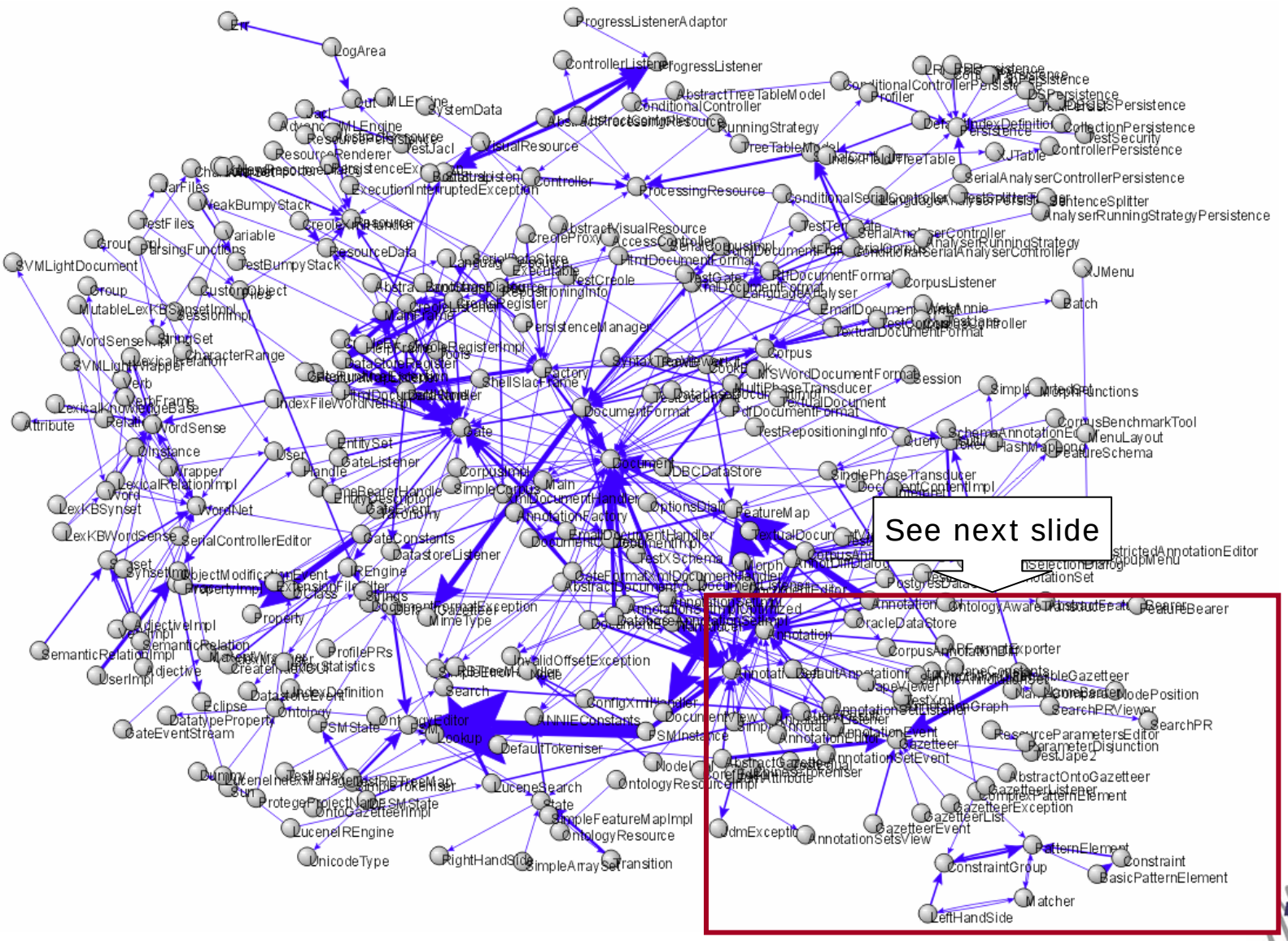
```
/** The MIME type of this format. */
 private MimeType mimeType = null;
```

```
/**
   * Find a DocumentFormat implementation that deals with a particular
   * MIME type, given that type.
   * @param  aGateDocument this document will receive as a feature
   *                       the associated Mime Type. The name of the feature is
   *                       MimeType and its value is in the format type/subtype
   * @param  mimeType the mime type that is given as input
   */
 static public DocumentFormat getDocumentFormat(gate.Document aGateDocument,
                                                MimeType mimeType){
```

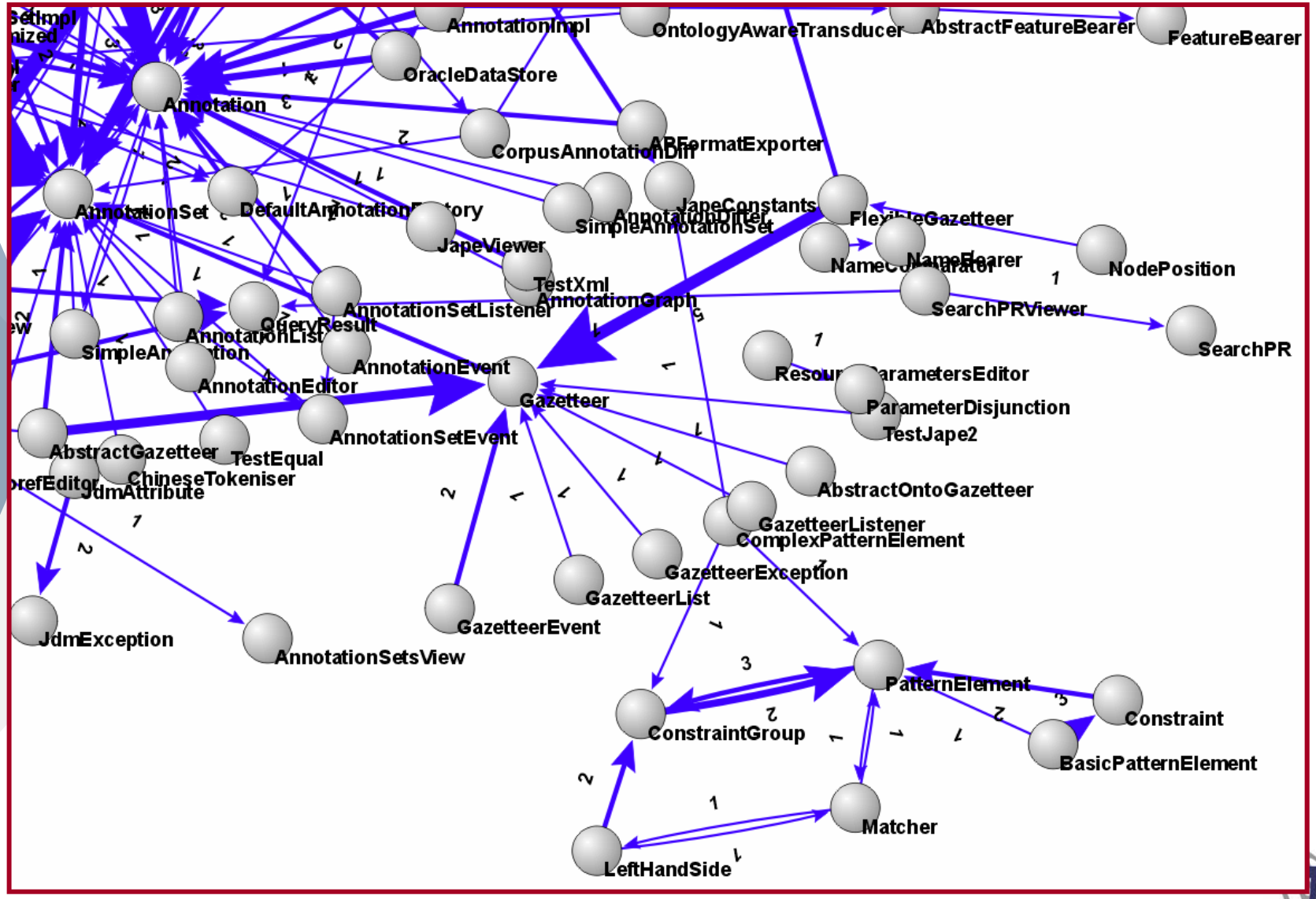
```
  } // getDocumentFormat(aGateDocument, MimeType)

} // class DocumentFormat
```



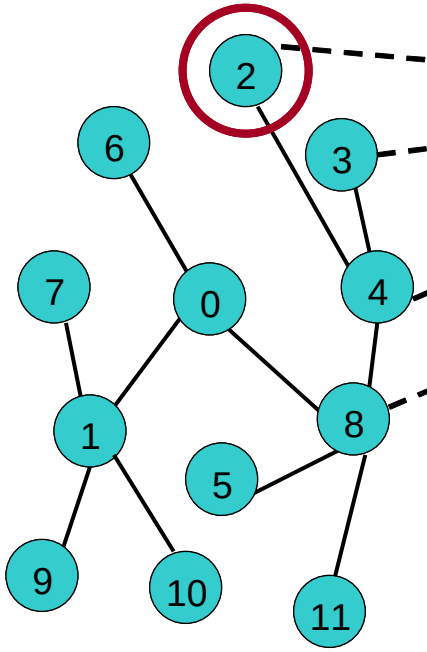
GATE Comment Reference Network



GATE Comment Reference Network

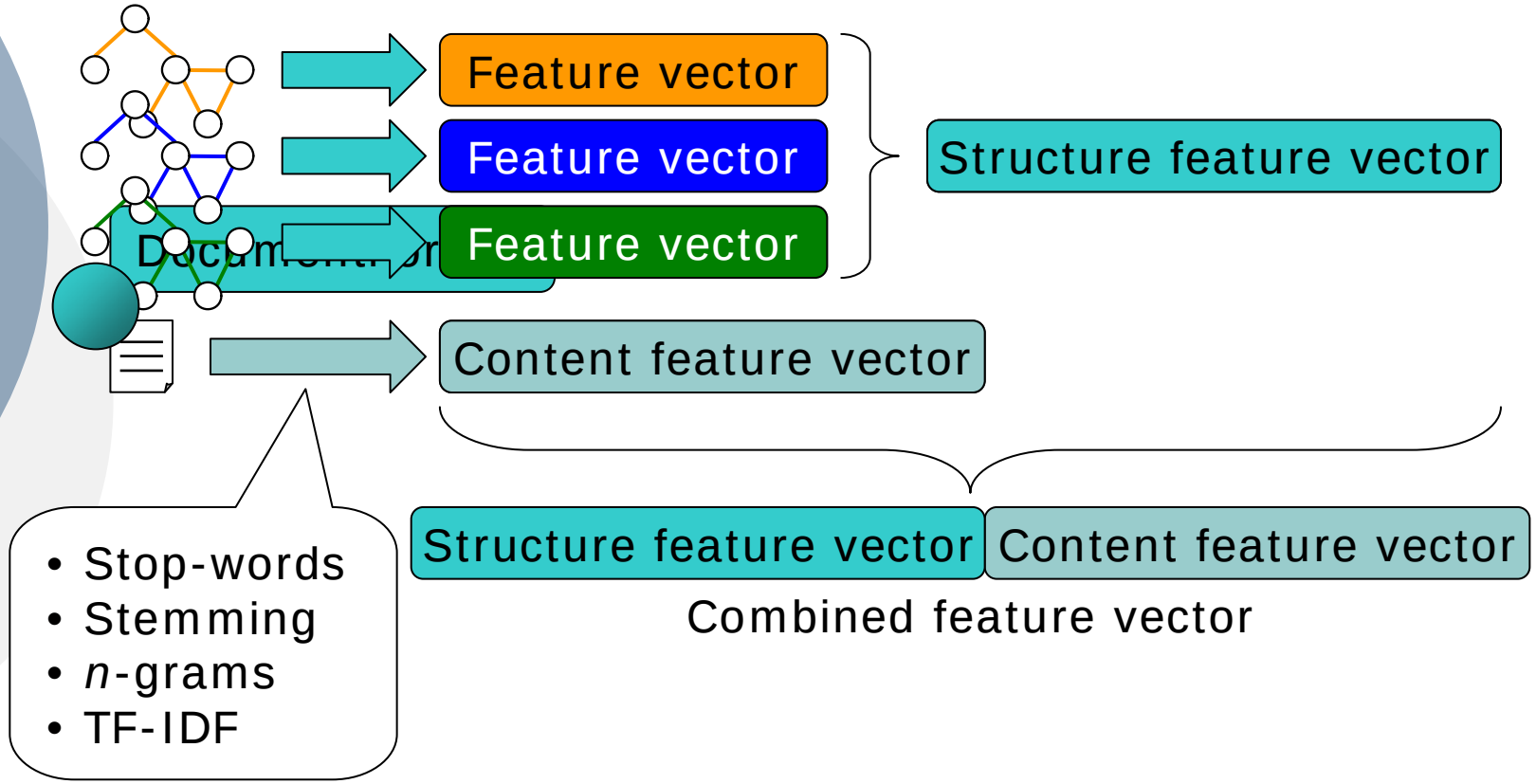


Transforming Networks into Feature Vectors



	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0.5			0.25	0.25	0.5	0.25	0.5	0.25	0.25	0.25
1	0.5	1					0.25	0.5	0.25	0.5	0.5	
2			1	0.25	0.5				0.25			
3			0.25	1	0.5				0.25			
4	0.25		0.5	0.5	1	0.25			0.5			0.25
5	0.25				0.25	1			0.5			0.25
6	0.5	0.25					1		0.25			
7	0.25	0.5						1		0.25		
8	0.5	0.25	0.25	0.25	0.5	0.5	0.25		1			0.5
9	0.25	0.5						0.25		1	0.25	
10		0.25								0.5	1	
11	0.25				0.25	0.25			0.5			1

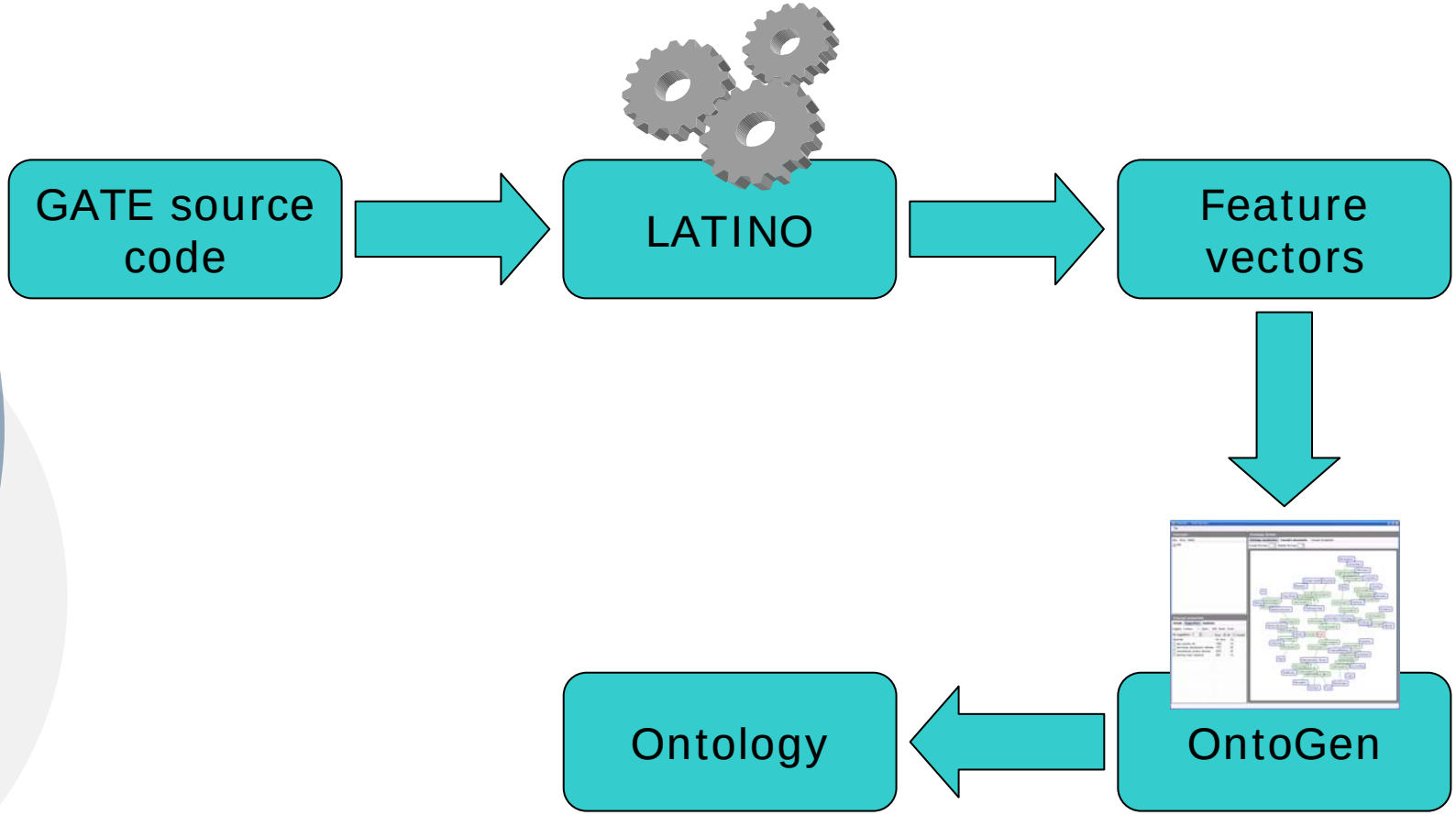
Combining Feature Vectors



LATINO & OntoGen Demo

- LATINO: Link analysis and text mining toolbox
  - Software being developed in the course of TAO WP 2
  - Data preprocessing, machine learning, and data visualization capabilities
- OntoGen
  - A system for *data-driven semi-automatic* ontology construction
  - SEKT technology (http://sekt-project.org)
  - Freely available at http://ontogen.ijs.si

LATINO & OntoGen Demo



OntoGen Demo

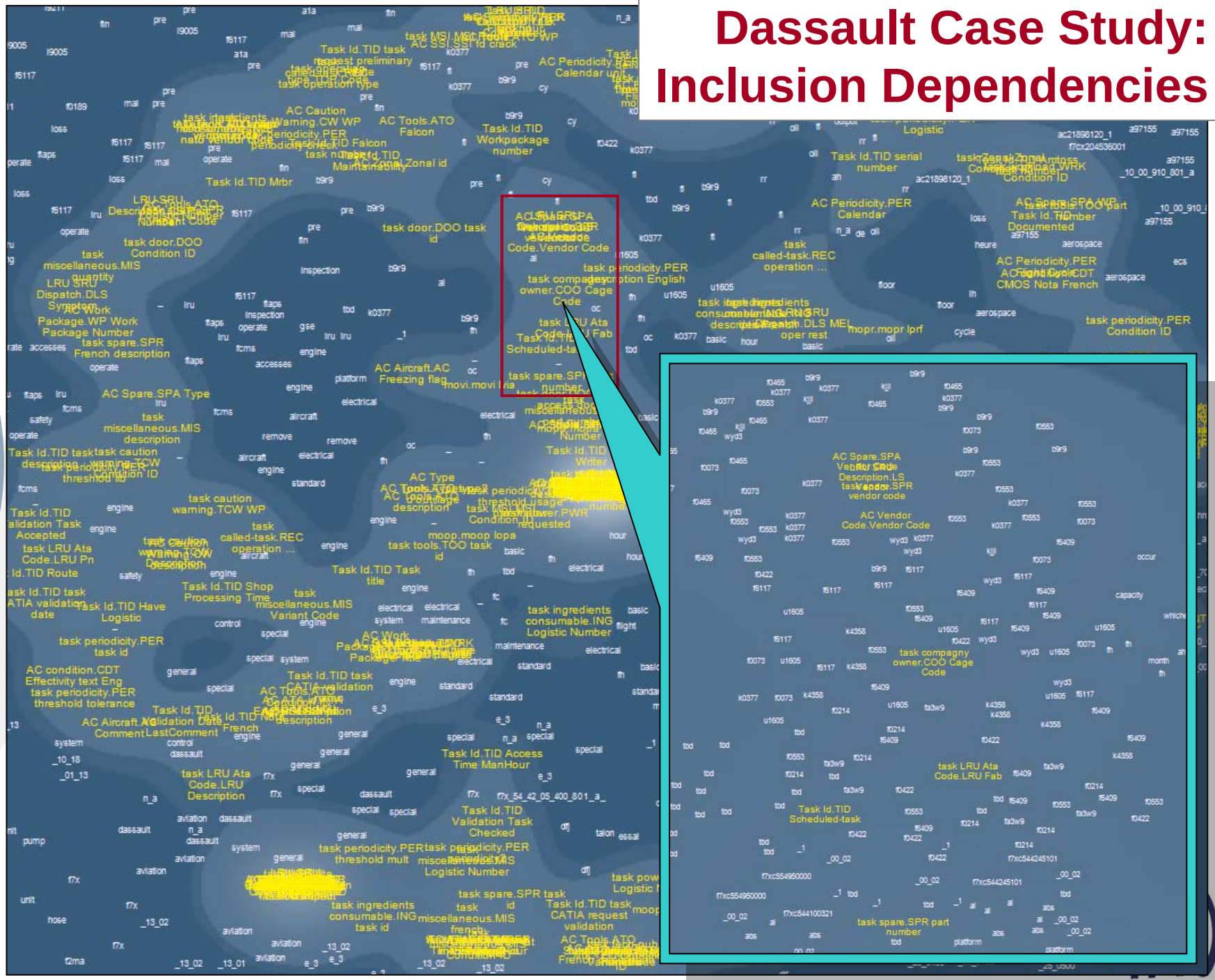
Dassault Case Study: Inclusion Dependencies

- Inclusion dependencies express subset-relationships between database tables and are thus important indicators of redundancy
- Discovery of ID important in the context of information integration
- Dassault Case Study
  - Problem: Dassault databases contain ID which should be taken into account when transitioning databases to ontologies
  - LATINO/OntoGen can help detect ID

Dassault Case Study: Inclusion Dependencies

- Dataset
  - The content of database tables in XML format
  - Ignore non-textual and empty table columns
- LATINO setting
  - Instances: columns (i.e. fields) in tables
  - Documents: concatenated values
  - Relations between instances:
    - Cosine similarity between documents
    - Similarity between sets of values
      - Jaccard, $|A \cap B| / |A \cup B|$
      - Alt., $|A \cap B| / \min\{|A|, |B|\}$
    - Edit distance (normalized) between column names

Dassault Case Study: Inclusion Dependencies



Dassault Case Study: Inclusion Dependencies

Candidates according to bag-of-words cosine sim

1.00 AC_Periodicity.PER_Aircraft : moop.moop_aircraft
1.00 AC_Periodicity.PER_Aircraft : mopa.mopa_kav
1.00 AC_Periodicity.PER_Aircraft : movi.movi_kav
1.00 AC_Periodicity.PER_Aircraft : AC_Zonal.Zonal_ac
1.00 AC_Tools.ATO_nato_vendor_code : task_miscellan to_vendo
...
0.99 AC_Tools.ATO_nato_vendor_code : task_ingredients_con nato_ve
0.99 task_ingredients_consumable.ING_nato_vendor_code : _Nato_v
0.99 task_ingredients_consumable.ING_nato_vendor_code
task_miscellaneous.MIS_nato_vendor_code
0.99 Task_Id.TID_task_owner : task_ingredients_consumable ndor_co
0.98 AC_Zonal.Zonal_ac : LRU_SRU_Description.LS_Aircraft
...
0.50 task_periodicity.PER_periodicity_usage_parameter2 : ts meter.US
0.50 task_periodicity.PER_threshold_usage_parameter : task eter.USP
0.49 Task_Id.TID_usage parameter : task_periodicity.PER_ _parame
0.48 mope.mope_kpe : task_periodicity.PER_threshold_tol us
0.48 mope.mope_kpe : task_periodicity.PER_periodicity_usage
...
0.21 AC_Vendor_Code.Vendor Code : task_LRU_Ata_Code LRU
0.21 AC_Zonal.Zonal_title : moid.moid_lida
0.21 task_compagny_owner.COO_Cage_Code : task_LRU_Ata b
0.21 Task_Id.TID_Task_Writer : task_miscellaneous.MIS_part
0.20 Task_Id.TID_Scheduled-task : task_spare.SPR_part_num

Task_Id.TID_task_owner	task_ingredients_consumable.ING_nato_vendor_code
F0189	
198	
F6117	
F6117	
F6117	F6117
F6117	
F6117	F6117
F6117	
F6117	
F6117	
F6117	F6117
F6117	
F6117	F6117
F6117	
F6117	
F6117	F6117
F6117	
F6117	
F6117	
F6117	
F6117	F6117
F6117	
K4358	
F0073	198
	198
F0073	198
F6117	I9005
F0189	
F6117	
F6117	

Conclusions and Future Work

- Plans for LATINO
  - (Recognized?) open-source architecture for text mining and link analysis
  - Build a user community, put up a Web site, training, promotion ...
  - Applications!
    - ... in case studies
    - ... in other EU projects
    - ... outside the context of EU projects
    - ... competing in data mining contests
- Future work
  - Implementation of a visualization tool similar to DocumentAtlas (required for setting the weights and exploring the semantic space)
  - Evaluation!
    - Can we solve problems introduced by case studies better if we use LATINO methodology rather than using standard text mining approach?
  - Continue the development of LATINO