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
Case 1: Regular Web service
Case 2: C++/Java source code
Case 3: Database
Case 4: ...

Intermediate data representation

Ontology

OL part works for all cases

Case-specific “adapters”
Application Mining

Intermediate data representation

Structured data = networks

Unstructured data = textual documents

Document network

= 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 ...
/** 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
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, HTMLDocumentFormat. 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 gate the GATEDocument this document will receive as a feature
 * @param mimeType the associated Mime Type. The name of the feature is
 * MimeType and its value is in the format type/mimetype
 * @param mimeType the mime type that is given an input
 * @return DocumentFormat getDocumentFormat(gate.Document gateDocument, 
 * MIMEType mimeType)
 */

// getDocumentFormat(gateDocument, MimeType)

// class DocumentFormat
Creating a Document Network

DocumentFormat.class

```java
/** 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. RtfDocumentFormat.
* RtfDocumentFormat, MpegDocumentFormat. These classes register themselves
* with a static method receiving 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 MimeTypes mimeType = null;

/** Find a DocumentFormat implementation that deals with a particular
* MIME type, given that type.
* @param mimeType the Mime Type the associated MimeType. The name of the feature is
* MimeType and its value is in the format type/mimeType
* @param mimeType the mime type that is given as input
* @return a DocumentFormat that can handle the input
*/
static public DocumentFormat getDocumentFormat(String document, String mimeType)
{
    return new DocumentFormat(document, mimeType); // class DocumentFormat
}
```
GATE Comment Reference Network
Transforming Networks into Feature Vectors

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>1</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
```
Combining Feature Vectors

- Stop-words
- Stemming
- $n$-grams
- TF-IDF
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
    - Freely available at [http://ontogen.ijs.si](http://ontogen.ijs.si)
  - Available at [http://ontogen.ijs.si](http://ontogen.ijs.si)
LATINO & OntoGen Demo

GATE source code → LATINO → Feature vectors → Ontology → OntoGen
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, \( \frac{|A \cap B|}{|A \cup B|} \)
  ◇ Alt., \( \frac{|A \cap B|}{\min\{|A|,|B|\}} \)
  ◇ Edit distance (normalized) between column names
Dassault Case Study: Inclusion Dependencies
Dassault Case Study: Inclusion Dependencies

Candidates according to bag-of-words cosine similarity:

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_miscellaneous.MIS_nato_vendor_code

... 
0.99 AC_Tools.ATO_nato_vendor_code : task_ingredients_consumable.ING_nato_vendor_code
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.ING_nato_vendor_code
0.98 AC_Zonal.Zonal_ac : LRU_SRU_Description.LS_Aircraft

... 
0.50 task_periodicity.PER_periodicity_usage_parameter2 : task_indexed.LRU_SRU_Description.US_LRU_SRU_Description
0.50 task_periodicity.PER_threshold_usage_parameter : task_indexed.LRU_SRU_Description.US_LRU_SRU_Description
0.49 Task_Id.TID_task_usage_parameter : task_periodicity.PER_threshold_usage_parameter
0.48 mope.mope_kpe : task_periodicity.PER_threshold_toll
0.48 mope.mope_kpe : task_periodicity.PER_periodicity_usage

... 
0.21 AC_Vendor_Code.Vendor Code : task_LRU_Ata_Code.LRU_Fab
0.21 AC_Zonal.Zonal_title : moid.moid_lida
0.21 task_compagny_owner.COO_Cage_Code : task_LRU_Ata_Code.LRU_Fab
0.21 Task_Id.TID_Task_Writer : task_miscellaneous.MIS_part_number
0.20 Task_Id.TID_Scheduled-task : task_spare.SPR_part_number

http://www.tao-project.eu
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