Searching over Public Administration Legal Documents using Ontologies

Diego Berrueta  Jose E. Labra  Luis Polo

CTIC Foundation, R & D Department
University of Oviedo, Spain
Motivation

General Aim

Features of public administration documents
- Lots of documents are published everyday
- Concrete domains: Law, administration, etc.

Poor effectiveness of syntactic approaches

Our approach:
- We applied Semantic web tools and ontologies in this domain
- A search tool with ontology based query expansion

General aim:
- Bridge the gap between public administration and citizens
BOPA (Boletín Oficial del Principado de Asturias)
- Official Journal of the Principality of Asturias, Spain
- Asturias:
  - Northern region of Spain
  - Approx. 1.1 million people
The BOPA contains every announcement related with Asturias public administration

- Laws, resolutions, public auctions, etc.
- General interest to every citizen
- Specialized jargon

The previous system:

- PDF and plain HTML
- Simple syntactic Search tool
- Chronological and by title

Some statistics:

- 14,314 articles in 23,100 pages (2005)
We could not interfere with the established process
  - We were allowed to work only with the generated data
Adding tags to each document manually = not practical
  - Not enough personal and lots of documents
  - Error prone
We defined ontologies for our problem domain

- A team of 3 domain experts was employed

Ontologies were developed in a modular way

- One upper legal & administrative Ontology
- Several domain ontologies
  - We defined micro-ontologies for several contexts
  - Micro-ontologies import the upper ontology
  - Examples: public-employment, subventions, etc.
Each concept has a synonym set (synset) associated

- In practice, we use 2 types of synsets:
  - Input synset: Terms used in common language
    - Allow to find a concept from a query term
  - Output synset: Legal and administrative terms
    - Allow to find a document that contains a concept

Example:
- Concept: **Holiday**
- Input synsets: “holiday”, “vacation”, “break”
- Output synsets: “holiday” “vacation”
Ontology Based Search

Search Process overview

1. Match terms with concepts
   - Terms: holiday, janitor

2. Spread activation

3. Obtain list of words and weights
   - Word          Weight
     - janitor   1.0
     - holiday   1.0
     - vacation  1.0
     - staff     0.5
     - collective agreement 0.5
     - work day  0.5
     - legal contract 0.75

4. Apply enriched query

Query: “holidays of janitors”

Results

Motivation
- Ontology based search
- Other services
- System development
- Conclusions
Step 1. Match terms to concepts

- Query sentence is parsed to obtain terms
  - Remove non-content words like “the”, “but”, etc.
  - Remove suffixes of words
- Terms are matched against input synsets of ontology concepts
- A context (micro-ontology) is determined for the query
  - If several concepts belong to different contexts, we take the common one
  - In case of several choices, the user can select one
- Search can be restricted to that context
We use the spreading activation algorithm.

It returns a weighted list of related concepts.

\[ w_{ij} = \text{weight of relation between nodes } i \text{ and } j \]

\[ \Delta = \text{set of activation nodes} \]

- Initialized to initial query concepts
- Implemented as a queue ordered by activation value

\[ \Theta = \text{set of output concepts} \]

\[ N_{\text{min}} = \text{Minimum activation level} \]

while \( \Delta \neq \emptyset \) and \( N_k > N_{\text{min}} \)

extract a node \( n_k \) from \( \Delta \)
add \( n_k \) to \( \Theta \)

For each node \( n_i \) such that \( w_{ki} > 0 \)

\[ N_i = N_i + w_{ki} N_k \]

Add \( n_i \) to \( \Delta \)

endWhile

return \( \Theta \)
Ontology Based Search

Step 2. Spreading Activation

Example of Spread activation

Output concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>janitor</td>
<td>1.0</td>
</tr>
<tr>
<td>holiday</td>
<td>1.0</td>
</tr>
<tr>
<td>staff</td>
<td>0.5</td>
</tr>
<tr>
<td>collective agreement</td>
<td>0.5</td>
</tr>
<tr>
<td>work day</td>
<td>0.5</td>
</tr>
<tr>
<td>legal contract</td>
<td>0.75</td>
</tr>
</tbody>
</table>
For each concept, we obtain the representative words
- The words are obtained from the output synsets

### Output concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>janitor</td>
<td>1.0</td>
</tr>
<tr>
<td>holiday</td>
<td>1.0</td>
</tr>
<tr>
<td>staff</td>
<td>0.5</td>
</tr>
<tr>
<td>collective agreement</td>
<td>0.5</td>
</tr>
<tr>
<td>work day</td>
<td>0.5</td>
</tr>
<tr>
<td>legal contract</td>
<td>0.75</td>
</tr>
</tbody>
</table>

### Word list

<table>
<thead>
<tr>
<th>Word</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>janitor</td>
<td>1.0</td>
</tr>
<tr>
<td>holiday</td>
<td>1.0</td>
</tr>
<tr>
<td>vacation</td>
<td>1.0</td>
</tr>
<tr>
<td>staff</td>
<td>0.5</td>
</tr>
<tr>
<td>collective agreement</td>
<td>0.5</td>
</tr>
<tr>
<td>work day</td>
<td>0.5</td>
</tr>
<tr>
<td>legal contract</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Ontology Based Search

Step 4. Apply syntactic search

- **Syntactic Search**
  - We used Apache Lucene search engine
  - Search expression:
    
    ```
    "janitor"^1 "holiday"^1 "vacation"^1 "staff"^.5
    ("collective agreement"~1)^.5 ("work day"~1)^.5
    ("legal contract"~1)^.75
    ```

- Search results are ordered according to relevance level
Other Services

Syntactic search

Motivation

Ontology based search

Other services

System development

Conclusions

CTIC Foundation

University of Oviedo, Spain
Other Services
Semantic search

Motivation
Ontology based search
Other services
System development
Conclusions

CTIC Foundation

University of Oviedo, Spain
Query subscription

- Users can subscribe to a query
- A notification can be sent when new results for that query appear
- Email and SMS alerts can be sent to registered users
User interface developed in SVG
Navigational representation of the Ontology concepts

Elementos:
- Concepto actual
- Conceptos más generales
- Conceptos más específicos
- Otras relaciones

Descripción:
Campo para las descripciones conceptuales.
### Explicación de los resultados de la búsqueda semántica

<table>
<thead>
<tr>
<th>Concepto</th>
<th>Explicación</th>
<th>Explicación en SVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portero</td>
<td>=&gt; Portero</td>
<td></td>
</tr>
<tr>
<td>Vacaciones</td>
<td>=&gt; Vacaciones</td>
<td></td>
</tr>
<tr>
<td>Contrato laboral</td>
<td>=&gt; Vacaciones =&gt; Contrato laboral</td>
<td></td>
</tr>
<tr>
<td>Elementos de un contrato laboral</td>
<td>=&gt; Vacaciones =&gt; Elementos de un contrato laboral</td>
<td></td>
</tr>
<tr>
<td>Convenio laboral</td>
<td>=&gt; Vacaciones =&gt; Elementos de un contrato laboral =&gt; Convenio laboral</td>
<td></td>
</tr>
<tr>
<td>Regulación empleado de fincas</td>
<td>=&gt; Vacaciones =&gt; Elementos de un contrato laboral =&gt; Regulación empleado de fincas</td>
<td></td>
</tr>
<tr>
<td>Empleado de fincas</td>
<td>=&gt; Portero =&gt; Empleado de fincas</td>
<td></td>
</tr>
<tr>
<td>Contratado</td>
<td>=&gt; Portero =&gt; Empleado de fincas =&gt; Contratado</td>
<td></td>
</tr>
<tr>
<td>Empleado de una comunidad</td>
<td>=&gt; Portero =&gt; Empleado de fincas =&gt; Empleado de una comunidad</td>
<td></td>
</tr>
<tr>
<td>Jornada laboral</td>
<td>=&gt; Vacaciones =&gt; Elementos de un contrato laboral =&gt; Jornada laboral</td>
<td></td>
</tr>
<tr>
<td>Trabajo de mantenimiento</td>
<td>=&gt; Portero =&gt; Trabajo de mantenimiento</td>
<td></td>
</tr>
<tr>
<td>Finca urbana</td>
<td>=&gt; Portero =&gt; Empleado de fincas =&gt; Finca urbana</td>
<td></td>
</tr>
<tr>
<td>Limpia</td>
<td>=&gt; Portero =&gt; Empleado de fincas =&gt; Limpia</td>
<td></td>
</tr>
<tr>
<td>Conserje</td>
<td>=&gt; Portero =&gt; Empleado de fincas =&gt; Conserje</td>
<td></td>
</tr>
<tr>
<td>Jardíniero</td>
<td>=&gt; Portero =&gt; Empleado de fincas =&gt; Jardínero</td>
<td></td>
</tr>
</tbody>
</table>

### Explicación

- **Motivation**: ontología basado en la búsqueda.
- **Ontology based search**: sistema de búsqueda semántica.
- **Other services**: desarrollo del sistema.
- **System development**: desarrollo del sistema.
- **Conclusions**: conclusiones del estudio.
J2EE application
- Data access tier
  - XML Native Database: XIndice
  - Relational database
  - Syntactical indexes managed by Lucene
  - Original web server where BOPA is still published
- Business tier
  - Several vertical subsystems (low coupling)
    - Search engines, syntactical analysis, ontology processing, etc.
  - Functionality exported through web services
- Main User interface developed with the Struts framework

Development methodology
- We used some agile development techniques
  - Pair programming
Recovering information from HTML

- A web crawler fetches HTML pages
  - The pages are not valid HTML
- JTidy transforms bad formed HTML to well formed XHTML
- A chain of XSLT transformations allows to extract information in XML format
  - We use a custom XML vocabulary
- Static HTML data is completed with database data
Interoperability has been a design goal

All the functionality can be accessed by web services

5 clients have been implemented

- Standard Web application front end
- VoiceXML interface
- Desktop .Net client
- Desktop Python client
- Java client using SWT libraries for Eclipse

We export a RSS channel which is updated daily with the contents of new bulletins

Bulletin contents are exported in HTML, PDF and RDF
At this moment, we are logging user activity and we collect different statistics
  - How many times the first result is selected, etc....

Developed framework for testing quality of results
  - Difficulties to define good search results
  - We apply it mainly for regression tests

We are planning to do medium scale tests with real users in short term
Conclusions and Future work

- **Practical application using a knowledge based approach**
  - The prototype has been adopted by the Public Administration department
    - It will be part of a production environment

- **Future work**
  - Extend the search to the whole public administration site
  - Practical evaluation of the approach
    - Usability, search results, etc.
  - Extend Context-Theory
    - Allow more flexibility in context definition
    - Relationship between contexts and spread-activated nodes
    - Allow users feedback (tags)
  - Ontology development and reusability
    - Incorporate other ontologies (DOLCE, SUMO, etc.)
    - Develop a general legal and administrative ontology
  - Digitization project of past documents
    - BOPA started to be published before 1900
The end