Making sense of Web content with Knowledge Patterns

Valentina Presutti
STLab ISTC-CNR, Italy
October 31\textsuperscript{st}, 2012 Linköping
Heterogeneity: strength and weakness of the Web

- The Web is the greatest source of open information ever existed.
- Such information is presented in a variety of contents – diverse format, conceptual model, semantics, evolution, etc.
Semantic Web as an empirical science

- A large set of realistic data, created by large communities of practice
- We can perform experiments on it
- Semantic Web can be founded as an empirical science, as a branch of web science
Objects of an empirical science

• An empirical science needs clear research objects,
  - e.g. cells, proteins, or membranes are types of research objects in different branches of biology.
• and develops procedures for making patterns emerge out of the research objects

• Web of data, social network data, bibliographical, musical, and multimedia data, RDFa, Microformats, etc., provide an empirical basis to the Semantic Web, and indirectly to knowledge engineering
Identifying, selecting, constructing patterns from SW research objects

Two main problems

1. The knowledge soup problem
   - The web of data is a knowledge soup because of the heterogeneous semantics of its datasets
   - Since people maintain and encode heterogeneous knowledge, how can formal knowledge be derived from the soup of triplified data?
Identifying, selecting, constructing patterns from SW research objects

Two main problems

• Two main problems

2. The knowledge boundary problem
   – How to establish the boundary of a set of triples that makes them meaningful i.e. relevant in context, so that they constitute a knowledge pattern?
   – How the very different types of data (e.g. natural language processing data, RDFa, database tables, etc.) that are used by Semantic Web techniques contribute to carve out that boundary?
How do we recognize situations?
Foreground and background

- People tend to remember things because they stick out from the background ("profiling")
  - but what makes a background as such?
- Expectations create scenarios
  - even things that are not there become part of the scenario if activated by an expectation
- Cf. Gestalt psychology (Köhler, Langacker, etc.)
Schema-based memory

- People tend to remember items that fit into a schema.
  - Things that are associated through some functional similarity (cf. Gibson’s affordances)
- Schemata seem to be learnt mostly inductively
  - blocks world, repeated verbalization of invariant scenes, peek-a-boo, etc. Cf. Deb Roy’s TED talk
- Schema similar to (conceptual) frame, script, knowledge pattern, etc.
Origin of modern frames and knowledge patterns

• «When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a Frame. This is a remembered framework to be adapted to fit reality by changing details as necessary ... a frame is a data-structure for representing a stereotyped situation» (Minsky 1975)

• Frames, schemas, scripts ... «These large-scale knowledge configurations supply top-down input for a wide range of communicative and interactive tasks ... the availability of global patterns of knowledge cuts down on non-determinacy enough to offset idiosyncratic bottom-up input that might otherwise be confusing» (Beaugrande, 1980)
Knowledge patterns (KP)

- We suggest the usage of frames as the primary research objects over the Semantic Web, as opposed to simple concepts or binary relations, and we call them knowledge patterns.

Knowledge patterns (KP) are cognitively and pragmatically relevant conceptual structures capturing a piece of generic ontological or procedural knowledge. A knowledge pattern logically formalizes conceptual structures as composed of concepts and relations between them.

- KPs are an abstraction of data structures like frames in linguistics and artificial intelligence, microformats and microdata in Web technologies, association rules and patterns in data mining technologies, and ontology design patterns.
Knowledge Pattern: manifestations

- Natural Language text: "Mary kissed John in the kitchen"
- RDF triples:
  - mykb.kiss_between_MI_and_B rdfs:type myont:KissingSituation
  - dbpedia:Michael_Jackson kisses dbpedia:Beyonce
  - mykb.kiss_between_MI_and_B myont:kissPerson dbpedia:Michael_Jackson
  - mykb.kiss_between_MI_and_B myont:kissPerson dbpedia:Beyonce
  - mykb.kiss_between_MI_and_B myont:place dbpedia:MTV

A database table:
- Person | Person | Event | Place
- Lora   | Chris  | kiss  | Rome

Image:
- Two children sitting in a boat with a dog.
How many KP?

We (STLab) are researching on

- collecting
  - reengineering
  - aligning
  - and using

knowledge patterns as keys for accessing
meaning of the Web
STLab research on KP

- Ontology design patterns
  - From linguistic frames, business models, database models, foundational and domain ontologies, etc.
- Pattern-based ontology design
- KP detection and discovery on linked data
- Frame-based machine reading and ontology learning
- KP-based knowledge extraction
  - Automatic entity typing, automatic link typing
- KP-based exploratory search
- ...
LET’S HAVE A CLOSER LOOK...
Top-down: expertise patterns

- Evidence that units of expertise are larger than what we have from average linked data triples, or ontology learning
  - Cf. cognitive scientist Dedre Gentner: "uniform relational representation is a hallmark of expertise"
  - We need to create expertise-oriented boundaries unifying multiple triples
  - "Competency questions" are used to link ontology design patterns to requirements:
    - Which objects take part in a certain event?
    - Which tasks should be executed in order to achieve a certain goal?
    - What’s the function of that artifact?
    - What norms are applicable to a certain case?
    - What inflammation is active in what body part with what morphology?
Ontology Design Patterns (ODP)

- A Content ODP is always associated with a General Use Case (GUC) expressed using Competency Questions (CQs)

- Example: InformationRealization

> What are the physical realizations of this information object?
> What information objects are realized by this physical object?
Layered pattern morphisms

- A logical design pattern describes a formal expression that can be exemplified, morphed, instantiated, and expressed in order to solve a domain modelling problem
  - owl:Class:_:x rdfs:subClassOf owl:Restriction:_:y
  - Inflammation rdfs:subClassOf (localizedIn some BodyPart)
  - Colitis rdfs:subClassOf (localizedIn some Colon)
  - John’s_colitis isLocalizedIn John’s_colon
  - “John’s colon is inflammed”, “John has got colitis”, “Colitis is the inflammation of colon”
ODP

- Collected at http://www.ontologydesignpatterns.org
- Currently: Logical ODPs, Content ODPs, Re-engineering ODPs, and Alignment ODPs
User-study

45 users, 3 sessions, controlled experiments
Task 1 without ODP, Task 2 with ODP

1. Are Content ODPs perceived as useful?
2. Are the ontologies constructed using Content ODPs better, in some modelling quality sense, than the ontologies constructed without patterns?
3. Are the tasks given to the participants solved faster when using Content ODPs?
4. How do participants use the Content ODPs provided, and what support would be beneficial?

Neither methodological nor tool support available in 2009

Eva Blomqvist, Aldo Gangemi, Valentina Presutti:
Ontology Evaluation

- Terminological Coverage

<table>
<thead>
<tr>
<th>Task</th>
<th>Sessions</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>1,2,3</td>
<td>83.1%</td>
</tr>
<tr>
<td>Task 2</td>
<td>1,2,3</td>
<td>69.1%</td>
</tr>
<tr>
<td>Task 2</td>
<td>1,3</td>
<td>74.3%</td>
</tr>
</tbody>
</table>

- Task Coverage

<table>
<thead>
<tr>
<th>Task</th>
<th>Sessions</th>
<th>Measure</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>1,2,3</td>
<td>Opt. supported CQs</td>
<td>30.0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supported CQs</td>
<td>65.2%</td>
</tr>
<tr>
<td>Task 2</td>
<td>1,2,3</td>
<td>Opt. supported CQs</td>
<td>39.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supported CQs</td>
<td>73.5%</td>
</tr>
<tr>
<td>Task 2</td>
<td>1,3</td>
<td>Opt. supported CQs</td>
<td>48.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supported CQs</td>
<td>88.3%</td>
</tr>
</tbody>
</table>
Ontology evaluation

Usability features of the ontologies (first task).

Usability features of the ontologies (second task).
Result overview

1. Are Content ODPs perceived as useful?
   – Yes!

2. Are the ontologies constructed using Content ODPs ‘better’, in some modelling quality sense?
   – Coverage of problem decreased (slower?) but major improvement in usability aspects, and fewer common mistakes

3. Are the tasks given to the participants solved faster when using Content ODPs?
   – Not really, rather slower (too little experience?)

4. How do participants use the Content ODPs provided, and what support would be beneficial?
   – How to find and select ODPs? How to reuse them? Tools?
XD Methodology and Tools (NTK)

- eXtreme Design (XD)
  - an agile method for developing ontologies with Content Patterns

- XD tool
  - a tool that supports XD method
  - released as both an Eclipse plugin and a NeOn Toolkit plugin
XD principles

- Customer involvement and feedback
- Customer stories, CQs and contextual statements

- CP reuse and modular design (ontology networks)
- Collaboration and integration
- Task-oriented design
- Test-driven design
- Pair design
XD Tools

Browse, search, and get content.

ODPs

Analyze your ontology against good practices and patterns.

Specialize, compose, annotate ODPs and ontologies.
User-study

35 users, 2 sessions, controlled experiments
Task 1 no ODP, Task 2 ODP+XD Tools, Task 3 + XD methodology

• Summary of research questions:
  1. Can we confirm the results from the previous study (Questions 1-4)? + How is modularity affected?
  2. Does XD Tools support the process of reusing CPs?
  3. Does the XD methodology support the process of reusing CPs, and does it affect any of the aspects from the previous study (e.g., time, quality)?

Eva Blomqvist, Valentina Presutti, Enrico Daga, Aldo Gangemi: Experimenting with eXtreme Design. EKAW 2010: 120-134
Results – Confirming previous conclusions?

1. Are CPs perceived as useful by the participants?
   - Confirmed – Increase for second session: Due to XD Tools?

2. Are the ontologies constructed using CPs ‘better’, in some modelling quality sense?
   - Coverage: Reduction of terminological coverage is no longer detected – Due to XD Tools?
   - Usability: Confirmed – Most prominent improvement!

3. Are the tasks solved faster when using CPs?
   - With tool support: no longer slower and less mistakes

4. What common modelling ‘mistakes’ can be identified, when not using patterns and when using CPs?
   - Decrease in occurrence of most frequent mistakes confirmed (44% average decrease) - Same types of mistakes
   - Two types of errors decrease significantly more than the others:
     • N-ary relations – decrease by 64%
     • Missing datatype properties – decrease by 46%
Do CPs increase the modularity of ontologies?
- Task 1: no ontologies are modularized
- Task 2: the ontologies contain on average 7.5 modules
- Conclusion: Since the participants choose to reuse the CPs as OWL-modules, rather than ideas for solutions, this inherently introduces modularity
Results: XD Tools

- The XD tool/plugin was useful for finding the patterns.
- The XD tool/plugin was useful for reusing the patterns.

The XD tool/plugin introduced too much overhead in the pattern reuse process.
Results XD: methodology

The XD methodology helped me to organize my work while modelling.

I already organized my work in a way similar to XD in the previous exercises...
Currently

• Continuing working on ODP-based ontology design
• Testing methodology and XD Tools extension
• Experiments to be conducted

Top-down: FrameNetLOD

- Bringing lexical resources on linked data (favor hybridization)
- Benefit from linking all lexical resources and have an homogenous more powerful one

Linking lexical knowledge to domain knowledge
- Linked data ground to lexical knowledge and textual documents

There are many issues related to the conversion of lexical resources – more specifically to semantic issues of FrameNet conversion. FrameNetLod provides:

- A method to solve those issues (supported by a tool)
- A conversion of FrameNet to RDF published as a dataset in the LOD
- A method to convert FrameNet data into knowledge patterns
FrameNet as LOD
FrameNet as LOD
FrameNet as ontologies
BOTTOM-UP: ACCESSING TEXTUAL KNOWLEDGE
Robust ontology learning (ROL)

- Fast and accurate NL to RDF/OWL transformation
  - Mid-to-strong variety of machine reading
- Good design quality of the resulting ontologies
  - more than entity + relations: aggregated data (frames, events) vs. sparse data
- Frame-based representation → Good design quality
  - [Coppola et al., ESWC-2009] shows that n-ary ontology design patterns (ODPs) can be easily derived from frames, and have equivalent conceptual expressivity (and have formal semantics in addition)
  - [Blomqvist, ISWC-2009] provides evidence that OL methods performances improve if the learning cycle is augmented with ODPs
Requirements for ROL on the Web

- Ability to map natural language (Web of documents, still the major part) to RDF/OWL representations (Web of data)
- Ability to capture accurate semantic structures (e.g. complex relations or frames)
- Easy adaptation to the principles of linked data publishing (IRI, links)
- Minimal computing time
Machine reading with FRED

The New York Times reported that John McCarthy died. He invented the programming language LISP.

FRED on STLab tools

FRED performs Robust Ontology Learning

(1) http://wit.istc.cnr.it/stlab-tools/fred/
FRED

- Based on Discourse Representation Theory and heuristics
- Produces OWL/RDF TBox and Abox
- Resolves entities on Linked Data
- Performs frame detection with a rule-based approach, with good performances
  - i.e. no training phase needed
- Is much faster as compared to other existing tools
- Quality of resulting ontologies?
  - Still to be rigorously evaluated but...
• We have demonstrated that this approach leads to promising results in large scale knowledge extraction
Tìpalo

- A FRED application
- Automatic typing of Wikipedia entities based on FRED
- Results are very good
- An indirect evaluation of FRED performances

Tipalo on STLab tool

Tipalo uses FRED and automatically assigns types to Wikipedia entities. Given a Wikipedia page URL, the tool returns an RDF graph composed of rdf:type, rdfs:subClassOf, owl:sameAs, and owl:equivalentTo statements providing typing information about the entity referred by the Wikipedia page.

How to use Tipalo as REST service

Online demo

Enter a Wikipedia page URI:
e.g., http://en.wikipedia.org/wiki/Wind_instrument

Examples

Wind instrument (http://en.wikipedia.org/wiki/Wind_instrument)
Paklo (http://en.wikipedia.org/wiki/Paklo)
Neutron star (http://en.wikipedia.org/wiki/Neutron_star)
Alter ego (http://en.wikipedia.org/wiki/Alter_ego)
Luperca (http://en.wikipedia.org/wiki/Luperca)
Chaise longue (http://en.wikipedia.org/wiki/Chaise_longue)
What does Tìpalo do?

- Goal: to guess the type of entities referred by Wikipedia, given their definition as provided by their Wikipedia page abstract.
How does it do it?
Performance of the individual components

<table>
<thead>
<tr>
<th>Component</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure (F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type selector</td>
<td>0.93</td>
<td>0.9</td>
<td>0.92</td>
</tr>
<tr>
<td>WSD (UKB)</td>
<td>0.86</td>
<td>0.82</td>
<td>0.84</td>
</tr>
<tr>
<td>WSD (most frequent sense)</td>
<td>0.77</td>
<td>0.73</td>
<td>0.75</td>
</tr>
<tr>
<td>Type matcher (Supersense)</td>
<td>0.73</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>Type matcher (DUL+/D0)</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Performance of the overall process

<table>
<thead>
<tr>
<th>Typing process</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure (F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordNet types</td>
<td>0.76</td>
<td>0.74</td>
<td>0.75</td>
</tr>
<tr>
<td>Supersenses</td>
<td>0.62</td>
<td>0.6</td>
<td>0.61</td>
</tr>
<tr>
<td>Dul+/D0</td>
<td>0.68</td>
<td>0.66</td>
<td>0.67</td>
</tr>
</tbody>
</table>
User-based evaluation

<table>
<thead>
<tr>
<th>Task</th>
<th>Type extraction</th>
<th>Taxonomy induction</th>
<th>WSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness</td>
<td>0.84</td>
<td>0.96</td>
<td>0.81</td>
</tr>
</tbody>
</table>

ONGOING...
Lègalo

• To guess the meaning hidden by hypertextual links, given the text surrounding anchors (href)
Adapting Tipalo process for guessing semantics of links
BOTTOM-UP: EXTRACTING KP FROM LINKED DATA
Schema extraction

1. Gather dataset stats
2. Extract paths (length 1 to 4)
3. Identify central types/properties
4. Extract emerging KPs
5. Build path clusters

Results

- A method for extracting the main knowledge patterns of a LD dataset
Path identification (length 3)

PREFIX mo : http://purl.org/ontology/mo/MusicArtist

mo:MusicArtist
mo:Record
mo:Track
mo:Playlist
mo:Signal
rdfs:Resource
rdfs:Literal
dc:title
mo:published_as
mo:recorded_as
mo:published_as

Path Element
Position 2

(Jamendo)

foaf:Document

PREFIX mo : http://purl.org/ontology/mo/MusicArtist
Centrality (types)

PREFIX mo : http://purl.org/ontology/mo/MusicArtist

mo:MusicArtist
mo:Record
mo:Track
mo:Playlist
mo:Signal

rdfs:Resource
rdfs:Literal

foaf:Document

mo:published_as
mo:track
mo:track_number
mo:license
mo:available_as

dc:title

(Jamendo)
Centrality (properties)

PREFIX mo : http://purl.org/ontology/mo/MusicArtist

mo:MusicArtist
mo:Track
mo:Record
mo:Playlist
mo:Signal
mo:published_as
mo:recorded_as
mo:available_as
mo:license
mo:track_number
rdfs:Literal
rdfs:Resource
foaf:Document
(Jamendo)

PREFIX mo : http://purl.org/ontology/mo/MusicArtist
Centrality in Jamendo

- Betweenness
- Frequency
Bottom-up: Encyclopedic Knowledge Patterns (EKP)

- Improving knowledge exploration and summarization by:
  - Empirically discovering invariances in conceptual organization of knowledge – encyclopedic knowledge patterns – from Wikipedia crowd-sourced page links
  - Understanding the most intuitive way of selecting relevant entities used to describe a given entity
  - Identifying the typical / atypical types of things that people use for describing other things
  - Enabling serendipitous search

Input data

- Wikipedia page links generate 107.9M triples
- Infobox-based triples are 13.6M, including data value triples (9.4M)
- “Unmapped” object value triples are only 7% of page links

Paths are used to discover Encyclopedic Knowledge Patterns. Such patterns should make it emerge the most typical types of things that the Wikipedia crowd uses to describe a resource of a given type.
An Encyclopedic Knowledge Pattern (EKP) is discovered from the paths emerging from Wikipedia page link invariances. They are represented as OWL2 ontologies.
Paths and indicators

- Emerging paths are stored in RDF according to the “Knowledge Architecture” vocabulary
  - Cf. our COLD2011 paper “Extracting core knowledge from linked data”
- Paths and types are associated with a set of indicators
nRes(dbpo:MusicalArtist)

Number of resources having type dbpo:MusicalArtist

PREFIX dbpo : http://dbpedia.org/ontology/

Anthony_Kiedis
Chad_Smith
Michael_Jackson
Jackie_Jackson
Paul_McCartney
John_Lennon
nSubjectRes(P_{i,j})

Dave_Grohl

Foo_Fighters

I

Michael_Jackson

Jackson_5

Jackie_Jackson

S_{i}

dbpo:MusicalArtist

dbpo:wikiPageWikiLink

O_{j}

dbpo:Band

Nirvana

Beatles

Paul_Mccartney

John_Lennon

PREFIX dbpo: http://dbpedia.org/ontology/
nSubjectRes(P_{i,j})

Number of distinct resources that participate in a path as subjects

PREFIX dbo: http://dbpedia.org/ontology/

Dave_Grohl
Foo_Fighters

Michael_Jackson
Jackson_5

Nirvana

Paul_McCartney
John_Lennon

Beatles

Jackie_Jackson
pathPopularity(P_{i,j}, S_i)

nSubjectRes(P_{i,j})/nRes(S_i)
Path Popularity distribution example

<table>
<thead>
<tr>
<th>Path</th>
<th>pathPopularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MusicFestival,Band]</td>
<td>82.33</td>
</tr>
<tr>
<td>[MusicFestival, MusicalArtist]</td>
<td>74.17</td>
</tr>
<tr>
<td>[MusicFestival, Country]</td>
<td>74.02</td>
</tr>
<tr>
<td>[MusicFestival, MusicGenre]</td>
<td>72.81</td>
</tr>
<tr>
<td>[MusicFestival, City]</td>
<td>38.37</td>
</tr>
<tr>
<td>[MusicFestival, AdministrativeRegion]</td>
<td>32.78</td>
</tr>
<tr>
<td>[MusicFestival, MusicFestival]</td>
<td>23.26</td>
</tr>
<tr>
<td>[MusicFestival, Album]</td>
<td>18.13</td>
</tr>
<tr>
<td>[MusicFestival, Film]</td>
<td>12.39</td>
</tr>
<tr>
<td>[MusicFestival, Stadium]</td>
<td>9.52</td>
</tr>
<tr>
<td>[MusicFestival, RadioStation]</td>
<td>9.52</td>
</tr>
<tr>
<td>[MusicFestival, Single]</td>
<td>8.76</td>
</tr>
<tr>
<td>[MusicFestival, Town]</td>
<td>8.61</td>
</tr>
<tr>
<td>[MusicFestival, Magazine]</td>
<td>8.46</td>
</tr>
<tr>
<td>[MusicFestival, Broadcast]</td>
<td>7.55</td>
</tr>
<tr>
<td>[MusicFestival, Newspaper]</td>
<td>6.95</td>
</tr>
<tr>
<td>[MusicFestival, TelevisionShow]</td>
<td>6.04</td>
</tr>
<tr>
<td>[MusicFestival, University]</td>
<td>5.74</td>
</tr>
<tr>
<td>[MusicFestival, Continent]</td>
<td>5.59</td>
</tr>
<tr>
<td>[MusicFestival, Comedian]</td>
<td>5.29</td>
</tr>
<tr>
<td>[MusicFestival, OfficeHolder]</td>
<td>4.98</td>
</tr>
<tr>
<td>[MusicFestival, Island]</td>
<td>4.98</td>
</tr>
</tbody>
</table>
Boundaries of EKPs

• An EKP($S_i$) is a set of paths, such that

• $P_{i,j} \in \text{EKP}(S_i) \iff \text{pathPopularity}(P_{i,j}, S_i) \geq t$

• $t$ is a threshold, under which a path is not included in an EKP

• How to get a good value for $t$?
## Boundary induction

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>For each path, calculate the path popularity</td>
</tr>
<tr>
<td>2.</td>
<td>For each subject type, get the 40 top-ranked path popularity values*</td>
</tr>
<tr>
<td>3.</td>
<td>Apply multiple correlation (Pearson ρ) between the paths of all subject types, by rank, and check for homogeneity of ranks across subject types</td>
</tr>
<tr>
<td>4.</td>
<td>For each of the 40 path popularity ranks, calculate its mean across all subject types</td>
</tr>
<tr>
<td>5.</td>
<td>Apply k-means clustering on the 40 ranks</td>
</tr>
<tr>
<td>6.</td>
<td>Decide threshold(s) based on k-means as well as other indicators (e.g. FrameNet roles distribution)</td>
</tr>
</tbody>
</table>

* 40 covers most “core” path popularity values, as well as many of the unusual ones.
k-means clustering on Path Popularity

Sample distribution of pathPopularity for DBpedia paths. The x-axis indicates how many paths (on average) are above a certain value $t$ for pathPopularity.

- 1 big cluster (4-cluster) with ranks below 18.18%
- 3 small clusters with ranks above 22.67%
- 1 alternative cluster (6-cluster) with ranks below 11.89%
What is the “agreement” between DBpedia and our sample users?

Average multiple correlation (Spearman $\rho$) between users’ assigned scores, and pathPopularity$_{DBpedia}$ based scores

$P_{i,j} \in EKP (S_i) \iff \text{pathPopularity}(P_{i,j}, S_i) \geq 11\%$

<table>
<thead>
<tr>
<th></th>
<th>DBpedia</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>0.893</td>
<td>Philosopher</td>
<td>0.661</td>
</tr>
<tr>
<td>Writer</td>
<td>0.748</td>
<td>Ambassador</td>
<td>0.655</td>
</tr>
<tr>
<td>Legislature</td>
<td>0.716</td>
<td>Album</td>
<td>0.871</td>
</tr>
<tr>
<td>Radio Station</td>
<td>0.772</td>
<td>Administrative Region</td>
<td>0.874</td>
</tr>
<tr>
<td>Country</td>
<td>0.665</td>
<td>Insect</td>
<td>0.624</td>
</tr>
<tr>
<td>Disease</td>
<td>0.824</td>
<td>Aircraft</td>
<td>0.664</td>
</tr>
</tbody>
</table>

Satisfactory precision
Aemoo

- [http://aemoo.org](http://aemoo.org) exploratory search application based on EKP

Semantic Web Challenge @ISWC 2011 – Short listed, 4th place
Comparing entities of the same type

Nicolas Sarkozy
Ronald Regan
Angela Merkel
Barack Obama
Silvio Berlusconi
Applying topic-sensitive EKP as lenses
Conclusions

- Cognitive science is quite explicit on what meaning is for humans: an activity of “framing reality for a purpose” (with broad sense of reality and purpose)
- Frames can be keys to that relational meaning
- The Semantic Web is now growing a lot of data: our tools are trying to make sense of them by using appropriate keys
- Empirical extraction and discovery of frames/knowledge patterns is feasible
- KP can be used for improving HCI e.g. in exploratory search