Pragmatic Ontology Evolution: Reconciling User Requirements and Application Performance

Francesco Osborne, Enrico Motta

@FraOsborne

Knowledge Media Institute
The Open University
United Kingdom
Ontology Evolution

Fundamental steps:

I. Detection of the need for the evolution.
II. Identification of candidate changes.
III. Validation and assessment of these changes.

Pragmatic needs:

- Compatibility with current workflows (e.g., similar # of concepts)
- Performance of relevant applications (e.g., recommender system)
- Explanation Capabilities for domain experts (e.g., trendy vs established concepts)
Ontology Evolution

Fundamental steps:

I. Detection of the need for the evolution.
II. Identification of candidate changes.
III. Validation and assessment of these changes.

Pragmatic needs:

- **Compatibility**
  - with current workflows (e.g., similar # of concepts)

- **Performance**
  - of relevant applications (e.g., recommender system)

- **Explanation Capabilities**
  - for domain experts (e.g., trendy vs established concepts)
Motivating Scenario: Evolving Springer Nature Internal Taxonomy I

Product Market Codes (PMC) is Springer Nature (SN) internal taxonomy of research field.

- Used for **tagging publications** with relevant topics.
- The Computer Science branch consists of **86 topics** on three levels.
- Outdated and too coarse grained.
  - e.g., Artificial Intelligence had no subtopics
Motivating Scenario: Evolving Springer Nature Internal Taxonomy II

The **Computer Science Ontology (CSO)** is a large-scale, automatically generated ontology of research areas.

- It was produced by applying the **Klink-2** algorithm on 16M publications.
- Current version includes about **14K topics** and **143K semantic relationship**.
- It gets **updated regularly** by running Klink-2 on recent corpora.
- It is available at [http://cso.kmi.open.ac.uk/](http://cso.kmi.open.ac.uk/)
The **Computer Science Ontology** (CSO) is a large-scale, automatically generated ontology of research areas.

- It was produced by applying the Klink-2 algorithm on 16M publications.
- Current version includes about 14K topics and 143K semantic relationships.
- It gets updated regularly by running Klink-2 on recent corpora.
- It is available at [http://cso.kmi.open.ac.uk](http://cso.kmi.open.ac.uk)

More information about CSO in the talk:

**The Computer Science Ontology:**
**A Large-Scale Taxonomy of Research Areas**

14:20 - 14:40
Fred Farr Forum
In this work we focused in particular on five PMC concepts:

- Artificial Intelligence (incl. Robotics)
- Computational Biology/Bioinformatics
- Systems and Data Security
- Software Engineering
- Computer Communication Networks.

We mapped them to nine CSO concepts obtaining 2,451 candidate concepts.

Pragmatic needs:

1. The resulting taxonomy should have about 120-200 concepts.
2. Support effectively tasks such as instance tagging, similarity computation, generation of recommendations, and clustering.
3. Provide explanation to users about the rationale for selecting concepts.
The Pragmatic Ontology Evolution (POE) framework is a novel approach for selecting a set of concepts able to produce an ontology:

1. **consistent** with the given requirements
2. parametrized with respect to a number of dimensions (e.g., topological considerations)
3. supporting effectively relevant computational tasks, such as:
   - instance tagging
   - similarity computation
   - generation of recommendations
   - data clustering
Pragmatic Ontology Evolution II

Input:
- Ontology
- Set of instances
- Set of candidate concepts
- Requirements
- Tasks
- Additional parameters (optionally)

1. Parameter Optimization

   Concept Ranking → Ontology Evaluation

   Best set of parameters and analytics

2. Recursive Concept Elimination

   Retrieves the $m$ best concepts.
   Generates ontologies missing one concept from the set.
   Weights the concepts according to ontology performance.
   Discard the worst concepts. Stops when reaches $n$ concepts.

Input:
- B Ontology
- B Set of instances
- B Set of (candidate concepts
- B Requirements
- B Tasks
- B Additional parameters (optionally)
POE executes a **grid search on the space of four parameters** and ranks the resulting ontologies according to their performance on the tasks.

It takes in consideration four dimensions:
- Temporal dimension.
- Internal versus external instances.
- Semantics.
- Structural considerations.

The output is the best ontology in the space of parameters.

<table>
<thead>
<tr>
<th>Task</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Tagging</td>
<td># of covered publications</td>
</tr>
<tr>
<td>Similarity Computation</td>
<td>$1 - \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (\cos(\theta_i, \theta_j) - \cos(f_i, f_j))^2}$</td>
</tr>
<tr>
<td>Generation of Recommendations</td>
<td>$\frac{1}{n} \sum_{i=1}^{n} \frac{\text{cov}(r_{ci}, r_{fi})}{\sigma_{rc_i} \sigma_{rf_i}}$</td>
</tr>
<tr>
<td>Clustering</td>
<td>$\frac{a_i + b_i}{n(2)}$</td>
</tr>
</tbody>
</table>

We evaluate on **multiple tasks** by averaging the metrics.
Parameter Optimization II

The results support domain experts in exploring the space of solutions.

Trendier topics works best
Recursive Concept Elimination (RCE)

RCE generates an ontology of $n$ concepts by applying 3 steps:

1. It produces an ontology with $m$ concepts (where $m > n$) using the best set of parameters found by the parameter optimization phase.
   - If no ontology of $m$ concepts complies with the requirements, these are temporarily relaxed.

2. It weights a concepts by generating a representation of the instances lacking that concept and evaluating it.

3. It discards the $j$ concepts with the smaller weights and returns to step 2, until it reaches $n$ concepts.

Differently from the typical strategy based on weighting single concepts, this solution considers concept synergy.
Evaluation I

Training dataset: 718 books (2012-2014)
Validation dataset: 500 books (2015-2016)

Four tasks:
1. Instance tagging
2. Similarity computation
3. Generation of recommendations
4. Clustering
**Evaluation II**

POE optimized for a task outperformed parameter optimization \((p=0.002)\) and the other baselines \((p=0.0004)\).

POE5 proved to be a good compromise by yielding a performance only marginally inferior to the version of POE optimized for the task \((p \geq 0.1)\).
Future Work

• Apply POE on **different kinds of ontologies** and tasks to derive useful guidelines on how to balance users and application needs.
• Address **more complex candidate changes**, involving different kinds of semantic relationships.
• On the technology transfer side, we will continue our **collaboration with Springer Nature** and provide further solutions to maintain and evolve their editorial ontologies.
Follow our work at http://skm.kmi.open.ac.uk/

Email: francesco.osborne@open.ac.uk
Twitter: FraOsborne
Site: people.kmi.open.ac.uk/francesco

More information about CSO:

The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas
14:20 - 14:40
Fred Farr Forum

See also...

The Computer Science Ontology
http://cso.kmi.open.ac.uk/

Smart Topic Miner
(SN application for tagging publication)
http://skm.kmi.open.ac.uk/stm/

Smart Book Recommender
(SN application for suggesting books)
http://skm.kmi.open.ac.uk/sbr/