SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation

By
Thong Chanchaem

Introduction

What is … ?

- Semantic Web: a vision of a future web of machine understandable document and data.
- XML, RDF and OWL: semantic web format.

Introduction (OWL example)

Contents

- Introduction
- Algorithms
- Summary

Introduction

OWL [2] is a set of XML elements and attributes, with standardized meaning, that are used to define term and their relationship.

OWL extends RDF Schema:

- OWL Class, equivalentProperty, sameIndividualAS ...
- RDF Schema SubClassOf, resource, ID ...

Introduction

TAP KB: a knowledge base that contains a broad range of lexical and taxonomic about popular object like: music, movie, author, place, etc.

- Browse the TAP KB
- Example of Places.rdf file
- Tap Activity Based Search
Goal

- To perform automated semantic tagging of large corpora.
- To introduce a new disambiguation algorithm to resolve ambiguities in a natural language corpus.
- To introduce the platform which different tagging applications can share.

How they do that?

- **SemTag** : an application written on the platform that perform automated semantic tagging of large corpora.
- **Seeker** : a platform for large-scale text analytics.
- **TBD** : a new algorithm for Taxonomy-Based Disambiguation.

SemTag

“The Chicago Bulls announced yesterday that Michael Jordan will…”

The `<resource ref=http://top.stanford.edu/Basketball Team_Bulls"-Chicago Bulls"/resource>announced yesterday that `<resource ref=http://top.stanford.edu/ Athletes/Jordan"-Michael">Michael Jordan</resource> will…”

SemTag

- **SemTag** uses TAP KB to build a web scale ontology.
- **SemTag** uses the concept of label bureaus from JPLS to obtain semantic annotation from the third party.

SemTag Architecture

Two fundamental categories of ambiguities

- Some labels appear at multiple locations in the TAP ontology.
- Some entries have labels that occur in contexts that have no representative in the taxonomy.
**Term definitions**

- O (ontology) is defined by four elements
  - C (Class)
  - S ⊆ C (subClass relation)
  - I ⊆ C (instances relation)
  - T ⊆ C (type relation)

- T (Taxonomy) is defined by three elements
  - V (a set of nodes)
  - r (root)
  - p : V → V

**Algorithm Sim**

\[ \text{Sim}(c, v) \]

\[
\begin{align*}
\text{Let } b &= \arg\max_i \left\{ f_k(c_i) \right\} \\
\text{if } b = \tau &\text{ return } 0 \\
\text{else } &\text{ return } 1
\end{align*}
\]

**Algorithm TBD**

\[ \text{TD}(c, v) \]

\[
\begin{align*}
\text{Let } u &\text{ be the nearest ancestor of } v \text{ with a measurement} \\
\text{if } 0.3 - \|u\| > 0.5 - \|c\| &\text{ return } 1 \\
\text{if } \|u\| \leq 0.5 &\text{ return } 1 \\
\text{else } &\text{ return } 0 \\
\text{else } &\text{ if } \|u\| > 0.5 \text{ return } \text{Sim}(c, u) \\
\text{else } &\text{ return } 1 - \text{Sim}(c, v)
\end{align*}
\]

**Results of SemTag**

They applied SemTag to set of 267 million pages producing 270G of dump data corresponding to 550 million labels in context.

Approximately 79% are judged to be on-topic, resulting in a final set of about 454 million spots, with accuracy around 82%.

**Nodes of TAP**

<table>
<thead>
<tr>
<th>Node</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAP</td>
<td>100%</td>
</tr>
<tr>
<td>Textography</td>
<td>75%</td>
</tr>
<tr>
<td>Information</td>
<td>72%</td>
</tr>
<tr>
<td>Technology</td>
<td>70%</td>
</tr>
<tr>
<td>Software</td>
<td>65%</td>
</tr>
<tr>
<td>Hardware</td>
<td>62%</td>
</tr>
<tr>
<td>Media</td>
<td>60%</td>
</tr>
<tr>
<td>Networking</td>
<td>58%</td>
</tr>
<tr>
<td>SoftwareStack</td>
<td>56%</td>
</tr>
<tr>
<td>OperatingSystem</td>
<td>55%</td>
</tr>
<tr>
<td>Storage</td>
<td>50%</td>
</tr>
<tr>
<td>Implementation</td>
<td>50%</td>
</tr>
<tr>
<td>Network</td>
<td>50%</td>
</tr>
<tr>
<td>Interoperability</td>
<td>50%</td>
</tr>
<tr>
<td>Middleware</td>
<td>50%</td>
</tr>
<tr>
<td>Cryptography</td>
<td>49%</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>49%</td>
</tr>
<tr>
<td>HardwareStack</td>
<td>48%</td>
</tr>
<tr>
<td>OperatingSystemStack</td>
<td>47%</td>
</tr>
<tr>
<td>StorageStack</td>
<td>45%</td>
</tr>
<tr>
<td>ImplementationStack</td>
<td>45%</td>
</tr>
<tr>
<td>NetworkStack</td>
<td>45%</td>
</tr>
<tr>
<td>InteroperabilityStack</td>
<td>45%</td>
</tr>
<tr>
<td>MiddlewareStack</td>
<td>45%</td>
</tr>
</tbody>
</table>

Nodes of TAP with percentage of topics containing in corresponding columns.
Design goal for Seeker

- Composability
- Modularity
- Extensibility
- Scalability
- Robustness

Infrastructure Components

- The Data Store
- The Index
- The Joiner

Architecture of the Seeker system

Advantage

- Other application can obtain semantic annotation from web-available database.
- They use both human and computer judgment to solve ambiguous data in their TBD algorithm.

Disadvantage

- The system requires a large amount of storage space to store data.

Future SemTag

- They will use some techniques to bootstrap from TAP to build much larger and richer ontologies in the future.
- Currently, SemTag uses RDF but in the future, SemTag will use advanced language as OWL.
Critical review

- This system writes the resulting annotations to the database which other mechanisms can obtain the data from. Can this concept work with any dynamic pages?
- They use only 11 volunteers to examine the selections. Is it enough? And Does the background of volunteers influence the judgment of label selecting?

Quiz

- What is the different between SemTag and Seeker?
- Why OWL is more advanced language than RDF?
- What does TBD do?
- Which ontology is used in the system?
- What makes SemTag and Seeker different form other applications?

References

[4] SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation