RIDAR
Relevant Internet Data Resource Identification

OntoCase
Case-Based Reasoning using Ontologies

Zoltán Balogh
Institute of Informatics
Slovak Academy of Sciences

The Knowledge Chain
RIDAR

Relevant Internet Data Resource identification

- **Requirement:** Identify which internet resources are relevant to a chosen domain
- **How?** RIDAR exploits existing search engines to retrieve URIs to relevant Internet resources based on users-supplied search terms or more complicated search expressions.
- **Result:** URIs of relevant internet resources
- Details about identified resources (URL, title, etc.) are stored into databases.

RIDAR Allows

- Plan search
- Management of app ID and license keys
- Integration of results from multiple search engines
- Storage of retrieved results into any “target” such as database or generic file through a common interface
- Reuse of already executed searches
- Manual input of resources

RIDAR Provides

- a common interface to Search Engines
- Stateless Web Service interface (Axis)
- Integrated APIs to Google, Yahoo and MSN (in progress)
Case-Based Reasoning

- Is the process of solving new problems based on the solutions of similar past problems.
- Like precedence in a legal trail
- We need statistically relevant data
- Phases: Retrieve, Reuse, Revise, Retain
- **Case** is a solution to given problem, where problem is a set of “feature – value” pairs and solution is a “result”
  \[ \{N, V, R\} \in C \]
- **Case Base** – set of all previous (historically) observed cases or set of generated a reused cases
- **Sample Problem**: What job positions are appropriate for me? 
  **Features**: int age, bool java_skill, bool linux_skill, int qualification, int employer
  **Result**: hired
  **Example Case**: age=30, java_skill=true, linux_skill=true, qualification=4
- **Sample Reuse**: hired : (age=53, java_skill=false, linux_skill=true, qualification=3)

Ontology-based Case-Based Reasoning

- **Prereq**: Resources are described in the ontology
- **Properties of instances are features**
- Features can be not only continuous and discrete but also:
  - Concepts (instance-of owl:Class) or
  - Instances (instance-of: owl:Thing) in the ontology
Case Retrieval through the use of Profiles

- we need to specify what to estimate (which Result concepts should be estimated) and which properties (WS invocation parameters and Resource properties) should be the part of the feature vector. For such purposes we are proposing to use a Profile concept.
Feature
- enables us to specify which property of a Resource to use as a feature during the retrieval process

WeightedFeature
- enables us to add weight to a feature (interval between <1-100>)

Feature Examples

Knowledge-based Approach for Service Performance Estimation
KAS-Design
OntoSims

- We introduced structural similarity of concepts in an ontology to refine the retrieval process of IBL.
- We have reached better results using Object Feature as when using just Numerical Feature.

\[
d(x,y) = \rho \frac{|a(x) \cap a(y)|}{|a(x)|} + (1 - \rho) \frac{|a(x) \cap a(y)|}{|a(y)|}
\]

Instance Based Learning

**Case Based Reasoning used as methodology**

1. Case Representation & Case Indexing
2. Case Retrieval
   - Nearest-neighbor retrieval (euclidean distance or other)
   - Locally (linear or polynomial) Weighted Regression
3. Case Adaptation
4. Case-Base Maintenance

**Instance-based learning deficiencies**

- Its major disadvantage is that it requires a large amount of historical data.
- Problem with qualitative features, where exact match of feature values must be made.
Thank You

Zoltan Balogh
balogh@savba.sk
Institute of Informatics, Bratislava