Consensus Ontologies

A Schema-Based Approach Combined with Inter-Ontology Reasoning

Laura Zavala

11/09/09

Consensus ontologies

A system that constructs a consensus ontology from numerous, independently designed ontologies.

Features

- The matching is carried out at the schema level.
- Requires no user intervention and is automated.
- No need for previous agreement on the semantics of the terminology used by each ontology.
- Integrates WordNet by using the JWNL API.
- Applies heuristic knowledge during linguistic matching.
- Reasons with additional relations during context matching.

The components

- Linguistic Matching
  - string and substring matching techniques
  - WordNet for preprocessing and to search for synonyms, antonyms, plurals, hypernyms, and hyponyms
- Contextual Matching
  - concept's property list
  - concept's relationship(s) with other concept(s)
- Reasoning Rules
  - based on the linguistic and contextual features
  - domain-independent rules, each regarding the relationship among ontology concepts

The algorithm

PUZZLE Algorithm – merge(G_1, G_2)
Input: Ontology G_1 and G_2
Output: Merged ontology G_2
Begin
    new location of G_1 's root = G_2 's root
    for each node C (except for the root) in G_1
        Parent(C) = C's parent set in G_1
        for each member p_i in Parent(C)
            p_j = new location of p_i in G_2
            relocate(C, p_j)
        end for
    end for
end

relocate(N_1, N_2)
Input: nodes N_1 and N_2
Output: the modified structure of N_2 according to information from N_1
begin
    if there exists any equivalentclass of N_1 in the child(ren) of N_2
        merge N_1 with it
    else if there exists any subclass of N_1 in the child(ren) of N_2
        Children(N_1) = set of such subclass(es)
        for each member c_i in Children(N_1)
            add links from N_2 to N_1 and from N_1 to c_i
            remove the link from N_2 to c_i
        end for
    else if there exists any superclass of N_1 in the child(ren) of N_2
        Parent(N_1) = set of such superclass(es)
        for each member p_i in Parent (N_1)
            recursively call relocate(N_1, p_i)
        end for
    else
        add a link from N_2 to N_1
    end if
end

Linguistic matching

Matching concept names

For any pair of ontology concepts C and C, their names N_C and N_C have the following mutually exclusive relationships, in terms of their linguistic features.

Relation	v_{name}	Candidate List	Example
anti-match	0		
exact-match	1	equivalent class	
sub-match	1	subclass	red car
super-match	1	superclass	
leading-match	lead	equivalent class	**act**ive
	substr		**act**or
	length		(3/6)
other	-		

Contextual matching

The context of an ontology concept C consists of two parts, its property list and its relationship(s) with other concept(s).

Property list matching

Considering the property lists, $P(C)$ and $P(C)$, of a pair of concepts C and C being matched, our goal is to calculate the similarity value *vProperty* between them:
$vProperty = wrequired * vrequired + wnon - required * vnon - required$

Relationships among concepts

- subclass, denoted by $\subseteq$
- superclass, denoted by $\supseteq$
- equivalentclass, denoted by $\equiv$
- sibling, denoted by $\approx$
- other, denoted by $\neq$

Reasoning rules: rule 1

Preconditions

$n_i(A) \equiv n_k(B)$ and $(n_i(A) \subseteq n_j(A)$ or $n_i(A) \supseteq n_j(A))$

Conclusion

$n_k(B) \subseteq n_j(A)$ or $n_k(B) \supseteq n_j(A)$

The superclass / subclass relationship of a class is transferable to its equivalent class(es)

Reasoning rules: rule 2

If two classes share the same parent(s), then their relationship is one of: equivalentclass, superclass, subclass, and sibling.

Preconditions

$n_{i1}(A) \supseteq n_{i2}(A)$ and $n_{k1}(B) \supseteq n_{k2}(B)$ and $n_{i1}(A) \equiv n_{k1}(B)$ and

1. $n_{i2}(A) \underset{\longleftrightarrow}{p} n_{k2}(B)$ and (the names of $n_{i2}(A)$ and $n_{k2}(B)$ have either an exactmatch, or a leading-match with $v_{name} = 0.8$)
2. $n_{i2}(A) \underset{\longrightarrow}{p} n_{k2}(B)$ and the name of $n_{k2}(B)$ is a sub-match of the name of $n_{i2}(A)$
3. $n_{i2}(A) \underset{\longleftarrow}{p} n_{k2}(B)$ and the name of $n_{k2}(B)$ is a super-match of the name of $n_{i2}(A)$
4. None of three above holds

Conclusion

$$n_{i2}(A) \equiv n_{k2}(B) \quad n_{i2}(A) \supseteq n_{k2}(B)$$
$$n_{i2}(A) \subseteq n_{k2}(B) \quad n_{i2}(A) \approx n_{k2}(B)$$

Reasoning rules: rule 3

If two classes have no direct relationship between them, we will refer to a third one, to find the semantic bridge between the original two.

Preconditions

$n_{i1}(A) \equiv n_{j1}(C)$ and $n_{j2}(C) \equiv n_{k2}(B)$ and $n_{k1}(B) \subseteq n_{k2}(B)$ and $n_{j1}(C) \subseteq n_{j2}(C)$ and

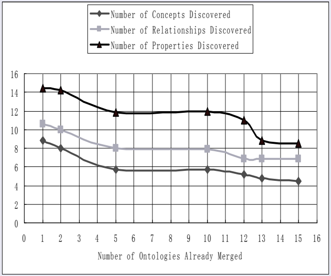
1. $n_{i1}(A) \underset{\longleftrightarrow}{p} n_{k1}(B)$ and (the names of $n_{i1}(A)$ and $n_{k1}(B)$ have either an exactmatch, or a leading-match with $v_{name} \geq 0.8$)
2. $n_{i1}(A) \underset{\longrightarrow}{p} n_{k1}(B)$ and the name of $n_{k1}(B)$ is a sub-match of the name of $n_{i1}(A)$
3. $n_{i1}(A) \underset{\longleftarrow}{p} n_{k1}(B)$ and the name of $n_{k1}(B)$ is a super-match of the name of $n_{i1}(A)$
4. None of three above holds

Conclusion

$$n_{i1}(A) \equiv n_{k1}(B) \qquad n_{i1}(A) \supseteq n_{k1}(B)$$
$$n_{i1}(A) \subseteq n_{k1}(B) \qquad n_{i1}(A) \approx n_{k1}(B)$$

Merging convergence

- 16 ontologies for "Building" domain
- 10 to 15 concepts
- 6 to 26 properties per concept



Observations

- As each additional ontology is merged into a consensus one, there should be fewer new items (concept, relationship, or property) added to the consensus.
- The resultant ontology represents a consensus model of a domain, but not necessarily a correct model.
- Our methodology is appropriate when there are a large number of small ontologies.
- Experiment with "Building" domain
    - PUZZLE removes redundant is-a links that are already specified by the transitivity of the superclasssubclass relationship.
    - The use of WordNet increases the accuracy. For instance, none of the original ontologies mentioned the relationship between the concepts Monument and Structure. However, PUZZLE found out that the concept Monument is a subclass of the concept Structure, which is quite reasonable and is an additional piece of information added to the merged ontology.