OntoSim - Ontology-based Similarity Determination of Concepts and Instances

Balogh Z., Budinská I.

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
balogh@savba.sk

Abstract. A tool OntoSim for determination of similarity between concepts and instances in the ontology is described in this article. Analysis of a existing method for concept comparison in ontology is made. Result of an implementation of a method for measuring concept similarity is evaluated. Experiments comprise generation of similarity matrixes for a sample test ontology.

Introduction

Recently ontologies are increasingly used for knowledge representation and modeling. They represent real-world semantics of data and relations among them. Ontologies enable to overcome heterogeneity by formal and explicit description of concepts and relations among them. Unfortunately, different data sources describing real-world entities, often do not relate to the same ontology concepts. In order to express the meaning of schema elements, they can be linked to different ontologies. This complicates searching for semantically equivalent schema elements. That is why it is important to have information about how much are concepts similar to each other. It also can support alignment and integration of ontologies.

Measurement of similarity between concepts and instances in ontology can be used for many different purposes by many tools that have been developing within the project NAZOU. As a theoretical base for the OntoSim tool a similarity graphs described in this paper is used. Except of the theoretical background, some implementation details are given in the section Implementation. Experimental results are described in section Experiment and some ideas for future work is given in Conclusion.

Similarity Graphs

Similar concepts are concepts that have much in common. To approximate this vague notion authors in [1] derive conceptual similarity using the notion of a „similarity graph": A similarity graph [2] is a subpart of the ontology represented as a graph with a subset of concepts as nodes and relations connecting these as edges. Reduced complexity is introduced by transforming formal conceptual reasoning into numerical concept similarity computation. The similarity between two concepts can thus be derived from a similarity graph covering these concepts.

For defining similarity, the objective is to derive a function sim(x; y) that measure degree of similarity proportional to how much the concepts x and y share or how close they are. Without loss of generality it is assumed that the function maps concepts into the unit interval:

$$sim(x, y) : C \times C \rightarrow [0, 1]$$

where C is the set of well-formed concepts and where sim(x; y) measure the degree to which y is similar to x. The extreme values sim(x; y) = 0 means not similar and sim(x; y) = 1 means fully similar. The latter may only be the case when x = y.

In [2] when restricting to similarity graphs the emphasis is put on the nodes „shared" by x and y. In [1] authors identified three major properties which guide the choice of similarity function:

1. *Generalization cost property* - the "cost" of generalization should be significantly higher than the cost of specialization;

By insisting on generalization cost property they have deduced that the similarity function cannot be symmetrical.

2. *Serialization cost property* - the "cost" of traversing edges should be lower when nodes are more specific;

The intuition for this property is that the similarity between for instance siblings on low levels in the ontology should be higher than the similarity between siblings close to the top as „Physical" and „Abstract".

3. *Specialization cost property* - further specialization implies reduced similarity.

In order to satisfy all the three above mentioned properties an alternative similarity function was proposed which is influenced by both specializations and generalizations but still not violates the only ultimate property above; the Generalization cost property:

$$sim(x,y) = \rho\frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1-\rho)\frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|}$$

Similarity between a given concept and all concepts C = {c1, ..., cin} in the ontology/database, thus to expand a concept c is computed as:

$$similar(c) = \sum_{i=1}^{CN} sim(c, c_i)/c$$

This can be viewed as a 2-dimensional matrix having size n2 where the value of the (i; j)'th entry is the similarity between the i'th and j'th concept.

Implementation

Data semantics are often represented via various ontology languages. Except traditional ontology languages, there is a number of ontology languages for semantic web, like SHOE, DAML, OWL. Effective mechanisms for similarity measure should cope with differences that are caused by different ontology languages. OntoSim is oriented on the OWL language, that is widely used within the community of semantic web and that is also used in the project NAZOU.

Implementation was executed using Java programming language and Jena Semantic development library. The objective of the implementation was to compute similarity matrix of a given sub-ontology. Such goal is performed by a sequence of the following steps:

a) Infer sub-classes of the main concept - for this purpose we used Jena generic inference engine, which was used to generate a new model of all sub-classes of a chosen main concept. The following rule was used as input for this:

```
[ (?C1 rdfs:subClassOf ?C2) (?C2 rdfs:subClassOf ?C3) ->
  (?C1 rdfs:subClassOf ?C3) ]
```

The above is a simple rule which infers that C1 is sub-class of C3 if there is a concept C2 for which holds that C1 is sub-class of C2 and C2 is sub-class of C3.

b) Load all inferred sub-classes of the main concept - in this step the new inferred relations stored in a new model needs to be connected to the original model, because the new model contains only new inferred relations. If the two models would not be connected, initial relations between concepts would be lost. After both models are connected, all concepts are retrieved - inferred and original as well.

c) Generate similarity matrix - similarity matrix is generated for all concept combinations. Therefore we generate a square matrix C x C, where C is a number of concepts in our sub-ontology. We use expression (1) to compute similarity among concepts; therefore we need to count shared nodes of two concepts. Having all inferred and original relations in one model we can retrieve all shared super-nodes using the following SPARQL query:

```
SELECT ?x
  WHERE {
    ns:a rdfs:subClassOf ?x .
    ns:b rdfs:subClassOf ?x .
  }
```

The above query retrieves all concepts **x** which are super-classes of both **ns:a** and **ns:b**.

Experiment

In the experimental phase we have used a sample ontology segment to compute similarities between individual Concepts in the ontology. We have created the following ontology:

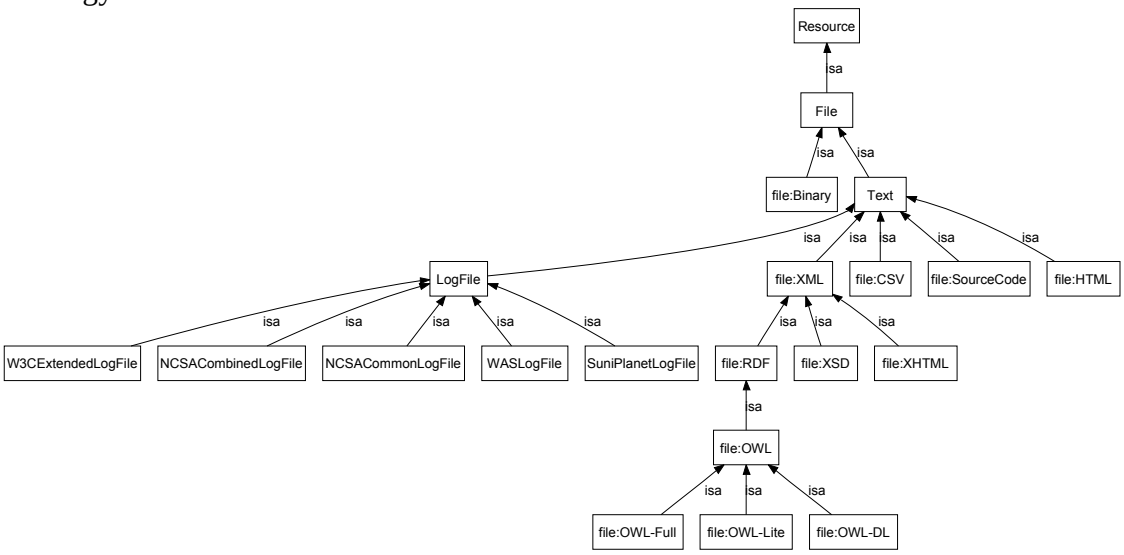

Figure 1: A sample ontology used in the experiment.

Similarity matrix generated from the above ontology is bellow divided into two tables:

		1	2	3	4	5	6	7	8	9	10
1	File	1,00	0,60	0,60	0,47	0,47	0,47	0,47	0,47	0,40	0,40
2	Binary	0,90	1,00	1,00	0,73	0,73	0,73	0,73	0,73	0,60	0,60
3	Text	0,90	1,00	1,00	0,73	0,73	0,73	0,73	0,73	0,60	0,60
4	LogFile	0,87	0,93	0,93	1,00	1,00	1,00	1,00	1,00	0,80	0,80
5	XML	0,87	0,93	0,93	1,00	1,00	1,00	1,00	1,00	0,80	0,80
6	CSV	0,87	0,93	0,93	1,00	1,00	1,00	1,00	1,00	0,80	0,80
7	SourceCode	0,87	0,93	0,93	1,00	1,00	1,00	1,00	1,00	0,80	0,80
8	HTML	0,87	0,93	0,93	1,00	1,00	1,00	1,00	1,00	0,80	0,80
9	W3CExtendedLogFile	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	1,00	1,00
10	NCSACombinedLogFile	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	1,00	1,00
11	NCSACommonLogFile	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	1,00	1,00
12	WASLogFile	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	1,00	1,00
13	SuniPlanetLogFile	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	1,00	1,00
14	RDF	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	0,75	0,75
15	XSD	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	0,75	0,75
16	XHTML	0,85	0,90	0,90	0,95	0,95	0,95	0,95	0,95	0,75	0,75
17	OWL	0,84	0,88	0,88	0,92	0,92	0,92	0,92	0,92	0,72	0,72
18	OWL-Full	0,83	0,87	0,87	0,90	0,90	0,90	0,90	0,90	0,70	0,70
19	OWL-Lite	0,83	0,87	0,87	0,90	0,90	0,90	0,90	0,90	0,70	0,70
20	OWL-DL	0,83	0,87	0,87	0,90	0,90	0,90	0,90	0,90	0,70	0,70

Table 1:First part of the similarity matrix computed in the experiment.

		11	12	13	14	15	16	17	18	19	20
1	File	0,40	0,40	0,40	0,40	0,40	0,40	0,36	0,33	0,33	0,33
2	Binary	0,60	0,60	0,60	0,60	0,60	0,60	0,52	0,47	0,47	0,47
3	Text	0,60	0,60	0,60	0,60	0,60	0,60	0,52	0,47	0,47	0,47
4	LogFile	0,80	0,80	0,80	0,80	0,80	0,80	0,68	0,60	0,60	0,60
5	XML	0,80	0,80	0,80	0,80	0,80	0,80	0,68	0,60	0,60	0,60
6	CSV	0,80	0,80	0,80	0,80	0,80	0,80	0,68	0,60	0,60	0,60
7	SourceCode	0,80	0,80	0,80	0,80	0,80	0,80	0,68	0,60	0,60	0,60
8	HTML	0,80	0,80	0,80	0,80	0,80	0,80	0,68	0,60	0,60	0,60
9	W3CExtendedLogFile	1,00	1,00	1,00	0,75	0,75	0,75	0,63	0,55	0,55	0,55
10	NCSACombinedLogFile	1,00	1,00	1,00	0,75	0,75	0,75	0,63	0,55	0,55	0,55
11	NCSACommonLogFile	1,00	1,00	1,00	0,75	0,75	0,75	0,63	0,55	0,55	0,55
12	WASLogFile	1,00	1,00	1,00	0,75	0,75	0,75	0,63	0,55	0,55	0,55
13	SuniPlanetLogFile	1,00	1,00	1,00	0,75	0,75	0,75	0,63	0,55	0,55	0,55
14	RDF	0,75	0,75	0,75	1,00	1,00	1,00	0,84	0,73	0,73	0,73
15	XSD	0,75	0,75	0,75	1,00	1,00	1,00	0,84	0,73	0,73	0,73
16	XHTML	0,75	0,75	0,75	1,00	1,00	1,00	0,84	0,73	0,73	0,73
17	OWL	0,72	0,72	0,72	0,96	0,96	0,96	1,00	0,87	0,87	0,87
18	OWL-Full	0,70	0,70	0,70	0,93	0,93	0,93	0,97	1,00	1,00	1,00
19	OWL-Lite	0,70	0,70	0,70	0,93	0,93	0,93	0,97	1,00	1,00	1,00
20	OWL-DL	0,70	0,70	0,70	0,93	0,93	0,93	0,97	1,00	1,00	1,00

Table 2: Second part of the similarity matrix computed in the experiment.

To get a similarity of concepts C1 with C2, concept C1 must be identified as a row and C2 as a column. The value identified on the intersection is the similarity of the concepts. It is important to note that similarity operation is not symmetric, therefore

sim(C1, C2) <> sim(C2, C1).

For example **sim(Text, File) = 0,90** and **sim(File, Text) = 0,60**.

The experiment has identified two main deficiencies of the proposed similarity computation, which we plan to improve in the future. First, the similarity values are dependent on the number of concepts in the similarity graph generated from the ontology used to describe a given domain. For this reason different similarities will be generated for two concepts for two different similarity graphs, while one similarity graph can just describe a problem domain in more details using more concepts. The other problem is, that concepts C1 and C2 which are direct sub-classes of another concept C3 are always 100% similar, i.e. sim(C1, C2) = 1,00. This is a problem in our view, because two concepts should be 100% similar only when they are identical, i.e.

sim(C1, C2) = 1,00 ↔ C1 = C2

We think this problem could be solved by assigning specialization and generalization weights to relations between concepts in the chosen similarity graph.

Conclusion

In this article we have described how we had implemented a tool called OntoSim using Jena Semantic Web development framework. We have used theoretical foundations of Similarity Graphs for our implementation. The results are satisfactory and the computed similarities consider serialization, specialization and generalization cost properties. There are many challenges for the future work we are willing to deal with including: computation of concept with multiple inheritances, refinement of specialization and generalization using weights (using for example a method as described in [4]) and extension of the presented method for similarity computation of instances.

Acknowledgements

This work is supported by projects K Wf-Grid EU RTD IST FP6-511385, NAZOU SPVV 1025/2004, RAPORT APVT-51-024604, VEGA No. 2/6103/6.

References

[1] Troels Andreasen, Henrik Bulskov, Rasmus Knappe: From Ontology over Similarity to Query Evaluation.
[2] Knappe, R., Bulskov, H. and Andreasen, T.: Similarity Graphs, LNAI 2871, pp. 668-672 in N. Zhong, Z.W. Ras, S. Tsumoto, E. Suzuki (Eds.): 14th International Symposium on Methodologies for Intelligent Systems, ISMIS 2003, Maebashi, Japan, October 28-31, 2003, Proceedings.
[3] Knappe, R., Bulskov, H. and Andreasen, T.: On Similarity Measures for Content-based Querying, pp. 400-403 in O. Kaynak et. al. (Eds.): 10th International Fuzzy Systems Association World Congress, IFSA 2003, Istanbul, Turkey, June 29-July 2, 2003, Proceedings.
[4] Michal Laclavik: Ontology and Agent based Approach for Knowledge Management; PhD Thesis submitted for the degree philosophiae doctor; Institute of Informatics, Slovak Academy of Sciences, field: Applied Informatics, June 2005, pp 82-83.