TOWARDS POLYGON-BASED SIMILARITY AGGREGATION IN ONTOLOGY MATCHING

Feiyu Lin and Kurt Sandkuhl
School of Engineering, Jönköping Universiy, Gjuterigatan 5, Jönköping, Sweden

Keywords: Polygon, similarity, aggregation, ontology matching.

Abstract: Due to an increased awareness of potential ontology applications in industry, public administration and academia, a growing number of ontologies are created by different organizations and individuals. Although these ontologies are developed for various application purposes and areas, they often contain overlapping information. In this context, it is necessary to find ways to integrate various ontologies and enable use of multiple ontologies. A number of concepts and approaches for ontology alignment and matching have been developed in this field. In this paper, we introduce our preliminary work on ontology matching using polygon-based similarity aggregation. The main ideas we contribute to the research field are (1) to aggregate the results of distance calculations between concepts in different ontologies by creating polygons for each ontology and (2) to compare the area of these polygons for deciding on similarity.

1 INTRODUCTION

When people or machines must communicate between themselves, they need a shared understanding of the same concepts. This finding from 19th century philosophy has influenced and been used in numerous research fields, like knowledge management, enterprise modeling or information systems and is one reason for the increasing use of semantic technologies.

Due to an increased awareness of potential ontology applications in industry, public administration and academia, a growing number of ontologies are created by different organizations and individuals. Although these ontologies are developed for various application purposes and areas, they often contain overlapping information. In this context, it is necessary to find ways to integrate various ontologies and enable cooperation. In this paper, we present our preliminary work on using polygon similarity for ontology matching.

2 RELATED WORK

There are many approaches that can be seen related to ontology matching. First we distinguish concepts about ontology mapping, matching and alignment, then we introduce different methods that can be used to calculate distance between concepts and we discuss related work.

2.1 Ontology Mapping / Matching / Alignment

The terms mapping, matching and alignment are frequently used in work about combining ontologies. Recent studies about combining ontologies (e.g. Noy and Musen, 2002) or (Keet, 2004)) distinguish between two principles approaches in this area: if the main objective is to combine two ontologies of the same subject area, this is denoted merging. In case the aim is to combine two ontologies from different subject areas, the term integration is used.

As a first phase in ontology merging and integration, the alignment of source ontologies is performed, aiming at identifying correspondences between the source ontologies. This process can be further divided into different steps such as mapping (i.e. finding equal parts in different source ontologies), matching (i.e. finding similar parts in the source ontologies) or finding translation rules between ontologies.

Lin F. and Sandkuhl K. (2007).
TOWARDS POLYGON-BASED SIMILARITY AGGREGATION IN ONTOLOGY MATCHING.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 391-395
DOI: 10.5220/0001271303910395

2.2 Distance Calculation

Different methods can be used to calculated distance between concepts in ontologies:

- String Similarity: (Cohen, 2003) has good survey of the different methods to calculate string distance.
- Synonyms (with the help of dictionary or thesaurus): Synonyms can help to solve the problem of using different terms in the ontologies for the same concept.
- Structure Similarity: This usually is based on *is-a* or *part-of* hierarchy of the ontology in the graph. Similarity flooding (Melnik, 2002) matching algorithm uses graphs to find corresponding nodes in the graphs based on a fix-point computation.
- Based on instances: Examples are GULE (Doan, 2002) or FCA-Merge (Stumme, 2001). GULE uses multiple machine learners and is exploiting information in concept instances and taxonomic structure of ontologies. FCA-Merge is a method for comparing ontologies that have a set of shared instances or a shared set of documents annotated with concepts from source ontologies.

2.3 Ontologies Matching Systems

There are some ontology matching systems available using some or all above methods. Examples are

- PROMPT (Noy, 2003) is a semi-automatically tool and a plug-in for the open-source ontology editor PROTÉGÉ (Protégé). It determines string similarity and analyzes the structure of ontology. It provides guidance for the user for merging ontologies. It suggests the possible mapping and determines the conflicts in the ontology and proposes solutions for theses conflicts.
- Chimaera (Chimaera) is a tool for the Ontolingua editor. It supports merging multiple ontologies and diagnosing individual or multiple ontologies. If string matches are found, the merge is done automatically, otherwise the user is prompted for further action.
- FOAM (Foam) is a tool to fully or semi-automatically align two or more OWL (OWL) ontologies. It is based on heuristics (similarity) of the individual entities (concepts, relations, and instances). These entities are compared using string similarity and SimSet for set comparisons.
- OLA (Euzenat, 2004.) takes care of all the possible characteristics of ontologies (i.e., terminological, structural and extensional). String similarity is used to calculate the labels' similarity. The structures constraints are considered during the matching.
- ASCO (Bach, 2004) uses as much available information in ontology as possible (e.g. concepts, relations, structure). It applies string similarity. TF/IDF is used for calculating similarity value between descriptions of the concepts or relations. WordNet (WordNet) is integrated to find synonyms. Structure matching is used for modifying or asserting the similarity of two concepts or relations.

3 SIMILARITY AGGREGATION BASED ON POLYGONS

Based on the methods discussed in 2.2, it is possible to calculate similarity expressed by the distances between concepts of ontologies. But even if we get the concepts' distances, we need an algorithm to aggregate the results of distance calculation in order to determine similarity on ontology level.

The main idea of our approach is to compare similarity between objects by comparing the area of polygons corresponding to each object, i.e. if polygons have exactly the same areas, similarity between the two objects represented by the polygons is maximal. An object in this context is a class within an OWL-ontology including properties and individuals. The creation of the polygon is based on a coordinate system with one half-axis for each attribute (i.e. property or individual) of the object. Furthermore, we use string similarity to determine the value that is used for representing the attribute on the corresponding half-axis. Connecting the values on the different axis with values on the adjacent axis creates the polygon representing the object.

3.1 Example Ontologies

We will use two simple ontologies about "match" in order to illustrate our approach (see Figure 1 and Figure 2).

In the ontology_1 (see Figure 1), object "match" has two object properties: "team" (related to "Idrott_förening" which has four subclasses: EgnahemsBK, HusqvarnaFF, Bankerydbasket and Sandabasket) and "plats" (related to "Rosenlunds"),

one datatype property “tid” which is string, and two subclasses: “Fotbollsmatch” and “Basketbollsmatch”. In the ontology_2 (see Figure 2), object “match” has two object properties: “ha_team” (related to “förening” which has four subclasses: Egnahems_BK, Husqvarna_FF, Bankeryd_basket and Sandabasket) and “ha_plats” (related to “Rosenlunds_IP”), one datatype property “ha_tid” which is date, and two subclasses: “Fotboll_match” and “Basketboll_match”.

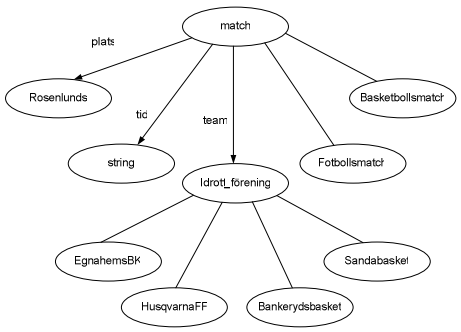

Figure 1: Ontology_1 relationship.

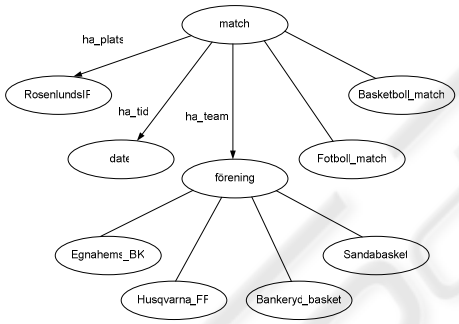

Figure 2: Ontology_2 relationship.

3.2 String Similarity Calculation

Cohen mentions that SoftTFIDF distance metrics shows the best result in experiments (Cohen, 2003).

Table 1: String distance in the ontologies (a).

Similarity	Ontology_1	Ontology_2
0.97	Rosenlunds	RosenlundsIP
0.71	Idrott_förening	förening
0.71	tid	ha_tid
0.71	team	ha_team
0.71	plats	ha_plats
0.64	Fotbollsmatch	fotboll_match
0.65	Basketbollsmatch	basketboll_match

We choose Jaro-WinklerTFIDF string-distance technique, which is based on SoftTFIDF and extended to use "soft" token-matching with the Jaro-Winkler distance metric.

After calculating Jaro-WinklerTFIDF string-distance using SecondString (SecondString) tool, we get the results shown in Table 1 and Table 2 for the example ontologies presented in section 3.1.

Table 2: String distance in the ontologies (b).

Similarity	Ontology_1	Ontology_2
1.0	Sandabasket	Sandabasket
0.68	EgnahemsBK	Egnahems_BK
0.68	HusqvarnaFF	Husqvarna_FF
0.64	Bankerydsbasket	Bankeryd_basket

3.3 Rules for Mapping Ontologies to Polygons

We are using the following rules to map an object, i.e. an OWL class with properties and individuals, to polygons:

1. **Choose the standard ontology.** In our example, we will take Ontology_1 as standard ontology. All the attributes of an object of the standard ontology, i.e. class properties and individuals, are marked as 1 unit when creating the polygon. In the following, the string distances are mapped to polygons as unit.
2. **Calculate string similarity between all attributes of an object in the standard ontology and the other ontology.** For our example ontologies, the result of this calculation is presented in Table 1 and Table 2.
3. **Use axes to present the objects’ attributes. Every axis can only add two object attributes. The points are added clockwise.** For example, from the Table 1 we know that “Rosenlunds” in ontology_1 is corresponding to “RosenlundsIP” in ontology_2. Y axis is used to present “Rosenlunds” with value 1 and “Idrott_förening” with value -1 (see Figure 3). Furthermore, Y axis is used to present “RosenlundsIP” with value 0.97 (see Table 1) and “förening” with value -0.71 (see Table1 and Figure 5).
4. **If a new axis is added, it halves the old axes.** For example, X axis is added by halving Y. Z axis is added by halving X and Y axes (see Figure 3 and 5).
5. **Skip mapping attributes on polygon which has no similarity.** For example, since there is

no string similarity between “string” and “date”, they are not appearing in Figure 3 and Figure 5. This kind of skipping nodes will cause problems. For example, if two ontologies have many attributes but just three or four attributes have 100% perfect match, the similarity between these two ontologies will be 1 if we just skip attributes which have no similarity.

6. **If the objects’ attribute has sub-attributes, introduce a new polygon based on currently axis (currently value divided by 2).** For example, if we compare the “match” between two ontologies, “Idrott_förening” and “förening” have subclasses. In ontology_1, “Sandabasket” is mapped to Y axis with value: 1 / 2 * 1= 0.5. Following the above rules 1, 2, 3, 4 and 5, we can get Figure 4. In the same way, in ontology_2, “Sandabasket” is mapped to Y axis with value: 0.71 / 2 * 1= 0.355 (Note: 0 .71 is the string distance between “Idrott_förening” and “förening” (see Figure 5 and table 1); 1 is the string distance between “Sandabasket” and “Sandabasket”.) This way we can also get Figure 6.

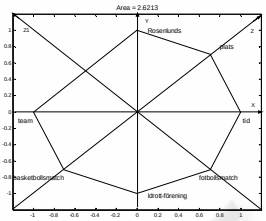

Figure 3: Ontology_1's polygon (a).

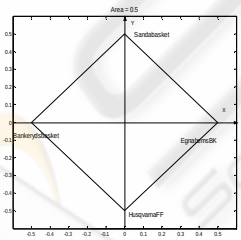

Figure 4: Ontology_1's polygon (b).

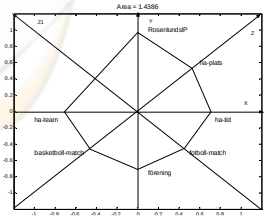

Figure 5: Ontology_2's polygon (a).

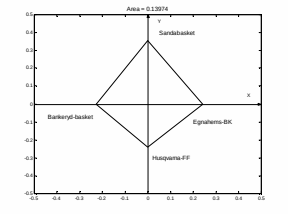

Figure 6: Ontology_2's polygon (b).

3.4 Calculation of Ontologies’ Similarity

In order to determine the similarity of the two ontologies, we calculate the polygons’ areas which were done using Matlab:

The area of polygon (a) for ontology_1 is 2.6213 (see Figure 3). The area of polygon (b) for ontology_1 is 0.5 (see Figure 4).

The area of polygon (a) for ontology_2 is 1.4386 (see Figure 5). The area of polygon (b) for ontology_2 is 0.13974 (see Figure 6).

We propose to calculate the area similarity by dividing the sum of the areas of all polygons related to ontology_1 by the sum of the areas of all polygons related to ontology_1.

Thus, the area similarity between ontology_1 and ontology_2 is:

$$SimOnto(a,b) = \frac{\sum_{i=1}^{n} area_{ai}}{\sum_{i=1}^{n} area_{bi}} \quad (1)$$

$$(1.4386 + 0.13974) / (2.6213 + 0.5) = 0.5057 \quad (2)$$

Since the area calculation is based on an area, i.e a two dimensional geometric figure, we should transform it to an expression reflecting just one dimension. The similarity between ontology_1 and ontology_2 then would be:

$$\sqrt{0.5057} = 0.7111 \quad (3)$$

In order to compare our approach with other approaches, we can for example calculate the similarity using mean values of the string distances or multiplying the string distances. This leads to the following result:

$$(0.97 + 0.71 + 0.71 + 0.71 + 0.71 + 0.64 + 0.65 + 1.0 + 0.68 + 0.68 + 0.64) / 11 = 0.7364 \quad (4)$$

Another way would be to calculate the similarity by multiplying the string distances:

$$0.97 * 0.71 * 0.71 * 0.71 * 0.71 * 0.64 * 0.65 * 1.0 * 0.68 * 0.68 * 0.64 = 0.0303 \quad (5)$$

We can see that the area result in this example is between the mean value and the multiplication value. Using mean value removes the high and low values' effect. Using multiplication zooms out the low values' effect. This is the reason we are proposing to use the area value.

4 SUMMARY AND FUTURE WORK

This paper presented a new approach for calculating similarity between objects and their different attributes based on polygons. The approach presented is still under development, i.e. the paper presents work in progress. Currently, a number of advantage and several shortcomings can be identified. Since objects can have many attributes, we consider polygons as suitable to represent these attributes.

- It is relatively easy to add or remove attributes in the polygon.
- It is a natural way to estimate object similarity by using shapes.
- It is easy to calculate similarity between polygons.

From the simple example presented in chapter 3 we can conclude that polygons are suitable for representing values derived from objects' attributes in an integrated manner. But there are still some problems which need to be solved in future work:

- The effect of the current approach of skipping nodes in the polygon with no similarity has to be investigated (see section 3.3). How to deal with this problem and improve the approach?
- How to add weights to the polygons reflecting the importance of attributes?
- How to combine our approach with other ontology matching methods, like synonyms, instance matching, structure matching, etc.
- Effects of choosing the standard ontology have to be investigated including use of the approach for more than two ontologies.
- Use of an alternative method to calculate polygon similarity instead of area. Currently, polygons with the same area have maximal similarity, even if they in reality are not identical.
- Comparison of string distance methods (e.g. Levenstein distance, Jaccard similarity…), to find the best string distance method for the polygon similarity.

The above problems will be investigated in future work. Furthermore, we plan to implement our polygon similarity approach and evaluate it in experiments. This will contribute important findings regarding the users' perception of accuracy of similarity calculation with our approach.

ACKNOWLEDGEMENTS

Part of this work was financed by the Hamrin Foundation (Hamrin Stiftelsen), project Media Information Logistics.

REFERENCES

Bach T.L., Dieng-kuntz R., Gandon F., 2004 *Ontology Matching: A Machine Learning Approach for building a corporate semantic web in a multi-communities organization.* In Proc. OFICEIS 2004, Porto (PT).

Chimaera. http://www.ksl.stanford.edu/software/chimaera/

Cohen W. W., Ravikumar P., and S. E. Fienberg, 2003 *A comparison of string distance metrics for name-matching tasks.* In Proceedings of the IJCAI-2003. http://citeseer.ist.psu.edu/cohen03comparison.html

Doan, A., Madhavan, J., Domingos, P., Halevy, A., 2002. *Learning to map between ontologies on the semantic web*. In: The Eleventh International WWW Conference. Hawaii, US.

Euzenat J., Valtchev P., 2004. *Similarity-based ontology alignment in OWL-lite*, In *Proc. 15th ECAI*, pp 333–337, Valencia (ES).

Foam http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/

Keet M., 2004. *Aspects of Ontology Integration*. (Unpublished, available at: http://www.dcs.napier.ac.uk/~cs203/AspectsOntologyIntegration.pdf)

Noy N., Musen, M., 2003. *The PROMPT suite:Interactive tools for ontology merging and mapping*. Technical report, SMI, Stanford University, CA, USA, http://citeseer.ist.psu.edu/noy03prompt.html

Noy N.F., Musen M.A., 2002. *Evaluating Ontology-Mapping Tools: Requirements and Experience.* In: Workshop on Evaluation of Ontology Tools at EKAW'02 (EON2002).

Owl. http://www.w3.org/TR/owl-features/

Protégé. http://protege.stanford.edu/

Melnik S., Molina-Garcia H., and Rahm E., 2002. *Similarity Flooding: A Versatile Graph Matching Algorithm*. In Proceedings of the International Conference on Data Engineering (ICDE).

SecondString. http://secondstring.sourceforge.net/

Stumme, G., Madche, A., 2001. *FCA-Merge: Bottom-up merging of ontologies.* In: 7th Intl. Conf.on Artificial Intelligence (IJCAI '01). Seattle, WA, pp. 225–230.

WordNet. http://wordnet.princeton.edu