The Knowledge Increase Estimation Framework for Ontology Integration on the Relation Level

Adrianna Kozierkiewicz-Hetmańska(✉) and Marcin Pietranik

Faculty of Computer Science and Management,
Wroclaw University of Science and Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{adrianna.kozierkiewicz,marcin.pietranik}@pwr.edu.pl

Abstract. The task of integration of sets of data or knowledge (regardless the choice of its representation) can be very daunting procedure, requiring a lot of computational resources and time. Authors claim that it is beneficial to develop a formal framework which could be used to estimate the profitability of the integration, ideally even before the integration even occurs. Therefore, a set of algorithms for such estimation of the increase of knowledge concerning relation level of ontology integration is proposed.

1 Introduction

One of the most common tasks related to knowledge management concerns its integration, which can be understood as a process of unification of a set of different and independent knowledge sources into one, consistent representation of the combined knowledge of the collective. This involves not only providing a summary of available information, but also resolving any conflicts which may entail inconsistencies and therefore, result with unreliable knowledge base. On the other hand, a new knowledge may appear as on outcome of a synergy - the unified, integrated collective knowledge may contain more information than a sum of its parts. To represent such knowledge a plethora of different methods and frameworks can be found in the literature. In our research we have focused on using ontologies as a knowledge representation. Their structure (according to [10]) can be expressed using a notion of "*ontology stack*" consisting of levels of concepts, relations and instances which express increasing level of abstraction of knowledge expressed within a particular ontology. The task of their integration can be formally defined as follows: *for given n ontologies* $O_1, O_2, ..., O_n$ *one should determine an ontology* O^* *which is the best representation of given input ontologies.*

This paper is devoted to the knowledge increase estimation framework serving as one of the quality assessment measures of ontology integration. This notion is related to answering the question about how much knowledge has been gained thanks to the performed integration and can be useful in a variety of applications like the one presented in [7]. Until now we have developed methods of estimating

N.T. Nguyen et al. (Eds.): ICCCI 2017, Part I, LNAI 10448, pp. 44–53, 2017.
DOI: 10.1007/978-3-319-67074-4_5

this knowledge increase during the integration of ontologies on the level of concepts and instances [5,6]. Due to the limitation of this paper only measures for concepts' relation level will be considered. To illustrate its usefulness we propose simple algorithms for ontology concepts' relations' integration, that are not a part of our framework.

Due to the limited space, the paper focuses only on the integration of concepts relations. The paper is organised as follows. Section 2 contains an overview of related works. Section 3 serves as an introduction to ontologies and basic notions used throughout the rest of the paper. In Sect. 4 the developed algorithms are described. This is followed by Sect. 5 that contains a variety of different use cases in which proposed measures may be useful. The last section is a summary and a brief description of authors' upcoming research plans.

2 Related Works

For assessing the integration process many authors [1,4] use popular measures like: completeness, precision, accuracy, consistency, relevance and reliability. However, the described functions have one, serious defect- all of them require the integration to be performed and only after it if completed they can be used to evaluate the obtained results. Other authors like [3] have considered ontology quality from the philosophical point of view where the data quality is defined and called as "fitness for use".

A more interesting solution has been presented in [13]. The overall model of ontology quality analysis has been proposed. Authors have defined the two types of metrics: schema and instance. Both are dedicated to the relation level of ontology. The relationship richness has reflected the diversity of relations and placement of relations in the ontology. Relationship richness classified to instance metric, reflecting how much of the properties in each class in the schema is actually being used at the instances level.

On the other hand, authors of [2] fit into the modern approach of treating everything as a service, by introducing a notion of Ontology as a Service (OaaS). To illustrate OaaS, they propose a sub-ontology extraction and merging, where a set of sub-ontologies are extracted from various input ontologies. Then extracted sub-ontologies are integrated to form a final ontology to be used by the user. However, authors do not propose any method of estimating a profitability of such process.

In [12] authors have presented a set of similarity measures between ontologies. Authors have distinguished two layers view of ontologies: lexical and conceptual. The relation overlap based on the geometric mean value of how similar their domain and range concepts are have been determined. This measure has reflected the accuracy that two relations match. Despite that authors have experimentally demonstrated the utility of their methods, the proposed measures are not able to estimate the potential knowledge increase during the integration process.

Lozano-Tello and Gómez-Pérez [8] have proposed the complex framework called Ontometric. Authors have defined a taxonomy of 160 characteristics, that

provides an outline able to choose and to compare existing ontologies. However, they have not been clearly presented and we suppose that Ontometric is not able to assess the growth of knowledge after adding a new ontology to the existing set.

Authors of [11] propose to approach ontology integration (also referred to as merging) as a task of ontology aggregation understood as a social choice. In other words, as a problem of aggregating the input of the procedure into an adequate collective decision. However, no estimation of the knowledge gained thanks to such collective approach has been given.

3 Basic Notions

In our framework a pair *(A, V)* denotes a real world, where A is a set of attributes that can be used to describe objects taken from some universe of discourse and V is a set of these attributes valuations. Formally $V = \bigcup_{a \in A} V_a$ where a domain of an attribute a is denoted as V_a. Ontology is a tuple:

$$O = (C, H, R^C, I, R^I) \tag{1}$$

where C is a set of concepts, H is concepts' hierarchy, R^C is a set of relations between concepts $R^C = \{r_1^C, r_2^C, ..., r_n^C\}$, $n \in N$, $r_i \subset C \times C$ for $i \in [1, n]$, I is a finite set of instances' identifiers and $R^I = \{r_1^I, r_2^I, ..., r_n^I\}$ denotes a set of relations between concepts' instances such that a relation r_j^C denotes a set describing possible connections between instances of some concepts from the set C and r_j^I are those connections actually materialised. In other words - relations from R^C define what objects can be connected with each other, while R^I defines what is connected. For example, in some ontology a set R^C may contain relations *is_sister* and *is_brother*, while R^I my contain definitions that John is a brother of Jane, and Jennifer is a sister of David.

Concepts taken from the set C are defined as $c = (id^c, A^c, V^c, I^c)$, where id^c is an identifier of a concept c, A^c is a set of its attributes, V^c is a set attributes domains (formally: $V^c = \bigcup_{a \in A^c} V_a$) and I^c is a set of particular concepts' instances. For short, we write $a \in c$ to denote that the attribute a belongs the the concept's c set of attributes A^c. An ontology is called *(A, V)-based* if the condition $\forall_{c \in C} ((A^c \subseteq A) \wedge (V^c \subseteq V))$

Concepts' instances are formally defined as a pair $i = (id^i, v_c^i)$, where id^i is its identifier and v_c^i is a function with a signature: $v_c^i : A^c \rightarrow V^c$. Referring to the consensus theory [9], the function v_c^i may by interpreted as a tuple of type A^c.

A set of instances from the base ontology definition (from the Eq. 1) is denoted below:

$$I = \bigcup_{c \in C} \{id^i | (id^i, v_c^i) \in I^c\} \tag{2}$$

we write $i \in c$ to denote a fact that the concept c contains an instance with an identifier i.

We define an auxiliary function Ins^{-1} that generates a set of concepts to which an instance with some identifier belongs. It has the signature $Ins^{-1} : I \rightarrow 2^C$ and is defined below:

$$Ins^{-1}(i) = \{c | c \in C \wedge i \in c\} \tag{3}$$

To simplify set operations we also define a set *Ins(c)* which contains only identifiers of instances assigned to concept *c*. Formally it can be defined as $Ins(c) = \{id^i | (id^i, v_c^i) \in I^c\}$.

L_s^R is a sublanguage of the sentence calculus and is used within a function that assigns semantics of relations from the set R^C. This function has a signature $S_R : R^C \rightarrow L_s^R$. As a consequence, we can define formal criteria for relationships between relations:

- *equivalency* between relations r and r' (denoted as $r \equiv r'$) occurs only if a sentence $S_R(r) \iff S_R(r')$ is a tautology
- a relation r' is more general than the relation r (denoted as $r' \leftarrow r$) if a sentence $S_R(r) \implies S_R(r')$ is a tautology
- *contradiction* between relations r and r' (denoted as $r \sim r'$)occurs only if a sentence $\neg(S_R(r) \wedge S_R(r'))$ is a tautology

The hierarchy of concepts (denoted in Eq. 1 as H) may be treated as a distinguished relation between concepts. Thus, $H \subset C \times C$. A pair of concepts $c_1 = (id^{c_1}, A^{c_1}, V^{c_1}, I^{c_1})$ and $c_2 = (id^{c_2}, A^{c_2}, V^{c_2}, I^{c_2})$ may be included within it (which will be denoted using a symbol $\leftarrow$), stating that c_2 is more general than c_1 ($c_2 \leftarrow c_1$), only if all of the following postulates are met:

1. $|A^{c_1}| \geq A^{c_2}$
2. $\forall a' \in A^{c_2} \exists a \in A^{c_1} : (a \equiv a') \vee (a' \leftarrow a)$
3. $Ins(c_1) \subseteq Ins(c_2)$

As previously stated, relations from the set R^C define which objects can be connected, while R^I defines what is actually connected. In our framework, to denote this fact, we will use the same index of relations taken from both sets. Therefore, a relation $r_j^I \in R_I$ contains only pairs of concepts' instances that are connected by a relation denoted as $r_j^C \in R^C$. A set of formal criteria that both sets must comply to is given below:

1. $r_j^I \subseteq \bigcup_{(c_1,c_2) \in r_j^C} (Ins(c_1) \times Ind(c_2))$
2. $(i_1, i_2) \in r_j^I \implies \exists (c_1, c_2) \in r_j^C : (c_1 \in Ins^{-1}(i_1)) \wedge (c_2 \in Ins^{-1}(i_2))$ which states that two instances may be in a relation with each other only if there is a relation connecting concepts they belong to
3. $(i_1, i_2) \in r_j^I \implies \neg \exists r_k^I \in R^I : ((i_1, i_2) \in r_k^I) \wedge (r_j^C \sim r_k^C)$ which describes that fact that two instances cannot be connected by two relations that have been defined as contradicting with each other using relations' semantics S_R
4. $(i_1, i_2) \in r_j^I \wedge \exists r_k^I \in R^I : r_k^C \leftarrow r_j^C \implies (i_1, i_2) \in r_k^I$ which states that if two instances are connected by some relation and there exists a more general relation, then these two instance are also connected by this relation

4 The Quantity of Knowledge on the Relations' Level of Ontologies

4.1 Overview of Integration Algorithms

The estimation of knowledge increase require information about how the two or more ontologies are integrated on relational level. In our work we assume the approach presented in Algorithm 1. It is conducted for two ontologies and any other new ontology can be iteratively added to the previous result. It is based on simple sum of parts of integrated ontologies and the only additional step is removing a redundant equivalent relations while preserving the knowledge about which concepts have been connected by discarded relations. At first, we also considered to remove relations that are a generalisation of other relations, but according to considerations from Sect. 3 discarding such relation may cause the loss of knowledge about connected concepts which may not meet requirements to participate in more specific relation. For example, coexisting more general relation "*is_family*" along with a relation "*is_mother*" should not entail its removal, due to the fact that it also expresses connections other than motherhood.

Algorithm 1. Concept relations integration

Require: Set of input ontologies: $O_1 = (C_1, H_1, R_1^C, I_1, R_1^I), O_2 = (C_2, H_2, R_2^C, I_2, R_2^I), ..., O_m = (C_m, H_m, R_m^C, I_m, R_m^I)$;

1: Set $R^* = \bigcup_{i=1}^{m} R_i^C$;
2: **for all** $(r, r') \in R^* \times R^*$ **do**
3: **if** $r \equiv r'$ **then**
4: $r = r \cup r'$;
5: $R^* = R^* \setminus \{r'\}$;
6: **end if**
7: **end for**

The integration of hierarchies in Algorithm 2 is different. It is not the integration of input hierarchies, but a process of generating a new taxonomy concepts that are a result of the ontology integration on concept level. The algorithm utilises criteria described in Sect. 3 and (as it will be further described in next section) it may create new relations that were not present in any of the input ontologies.

4.2 Algorithms for Knowledge Increase Estimation

The Algorithm 3 contains a procedure of calculating knowledge increase gained thanks to the integration of relations between concepts. It consists of three main steps, first of which being calculating the increase of knowledge coming from broadening the scope of equivalent relations. This situation refers to the fact that two equivalent relations may contain different pairs of concepts, and the

Algorithm 2. Hierarchy integration

Require: The integrated ontology $O^* = (C^*, H^*, R^{C^*}, I^*, R^{I^*})$ created from a set of input ontologies: $O_1 = (C_1, H_1, R_1^C, I_1, R_1^I), O_2 = (C_2, H_2, R_2^C, I_2, R_2^I), ..., O_m = (C_m, H_m, R_m^C, I_m, R_m^I)$;

Set $H^* = \phi$
for all $(c, c') \in C^* \times C^*$ **do**
  if $(c \leftarrow c')$ **then**
    $H^* = H^* \cup \{(c, c')\}$;
  end if
end for

eventual value of the increase of knowledge should reflect such supplementation. Second part of the algorithm concerns the integration of two relations, one being more general than the other. A naive approach would discard such relation, but this could entail a potential loss of knowledge because not all of the concepts in broader relation could participate in more specific interaction (e.g. not all parenting is a maternity). The last part reflects the situation in which two relations have nothing in common with each other, therefore the increase of knowledge can be maximal.

Algorithm 3. Knowledge increase during relations' integration

Require: A set of input ontologies: $O_1 = (C_1, H_1, R_1^C, I_1, R_1^I), O_2 = (C_2, H_2, R_2^C, I_2, R_2^I), ..., O_m = (C_m, H_m, R_m^C, I_m, R_m^I)$;

Set $R^* = \bigcup_{i=1}^{m} R_i^C$;
Set $R_U = R^* \times R^*$;
Set $\omega = |R^*|$;
Set $\Delta_R = 0$;
for all $(r, r') \in R_U$ **do**
  if $r \neq r'$ **then**
    if $r \equiv r'$ **then**
      $\Delta_R = \Delta_R + (1 - \frac{|r \cap r'|}{|r \cup r'|})$;
      $R_U = R_U \setminus \{(r', r)\}$
      $\omega = \omega - 1$
    else if $r \leftarrow r'$ **then**
      $\Delta_R = \Delta_R + \frac{|r \cap r'|}{|r|}$;
    else
      $\Delta_R = \Delta_R + 1$;
    end if
  end if
end for
return $\frac{\Delta_R}{\omega}$

Due to the fact that hierarchies are a specific kind of relations we claim that the increase of knowledge during the integration of ontologies should be

calculated separately using Algorithm 4. Equation 1 states that hierarchies are subsets of the Cartesian product of sets of concepts, so in the first step the algorithm checks if any of the integrated taxonomies are entirely included in the other one. If this is the case then the knowledge increase coming from origin ontologies is equal to 0. Otherwise the algorithm calculates ordinary Jaccard distance. This serves as an indication of how much knowledge has been gained thanks to strict integration of two ontologies and is denotes as δ_H^-. Due to the fact that the integration of hierarchies may result in new connections between concepts (utilising criteria from Sect. 3) the algorithm should handle such situation, because it may highly influence the final result. This is done by calculating the value δ_H^+ in the penultimate step of the algorithm. The final result is a simple sum of δ_H^- and δ_H^+. Obviously, the final value may be higher than 1 which represents the fact that the completely new knowledge (that has not existed in the partial ontologies) has been created as a result of the integration. This situation is discussed further in the next section of the article.

Algorithm 4. Knowledge increase during hierarchy integration

Require: The integrated ontology $O^* = (C^*, H^*, R^{C^*}, I^*, R^{I^*})$ created from two input ontologies: $O_1 = (C_1, H_1, R_1^C, I_1, R_1^I), O_2 = (C_2, H_2, R_2^C, I_2, R_2^I)$;

1: **if** $H_1 \subseteq H_2 \vee H_2 \subseteq H_1$ **then**
2: $\quad \delta_H^- = 0$;
3: **else**
4: $\quad \delta_H^- = 1 - \frac{|H_1 \cap H_2|}{|H_1 \cup H_2|}$;
5: **end if**
6: $\delta_H^+ = \frac{|H^* \setminus (H_1 \cup H_2)|}{|H_1 \cup H_2|}$
7: $\delta_H = \delta_H^+ + \delta_H^-$
8: **return** δ_H

5 Uses Case Scenarios for Hierarchy and Relation Integration

5.1 Hierarchy Integration

Let us illustrate by simple examples how the degree to which the knowledge increases is calculated in case of hierarchy integration (see Fig. 1). In the first case (Fig. 1A) the integrated ontologies are quite different. The fact that for these two ontologies any of the hierarchies is included in the other one entails that $\delta_H^- = 1$. After the integration, any of the new hierarchies is added and any of the old hierarchies is not replaced or removed. Therefore, $\delta_H^+ = 0$ because of the cardinality of a set $H* = H_1 \cup H_2$. Eventually, we obtain $\delta_H = 1$ and we can say that during the integration process on the relation level we doubled the knowledge we had.

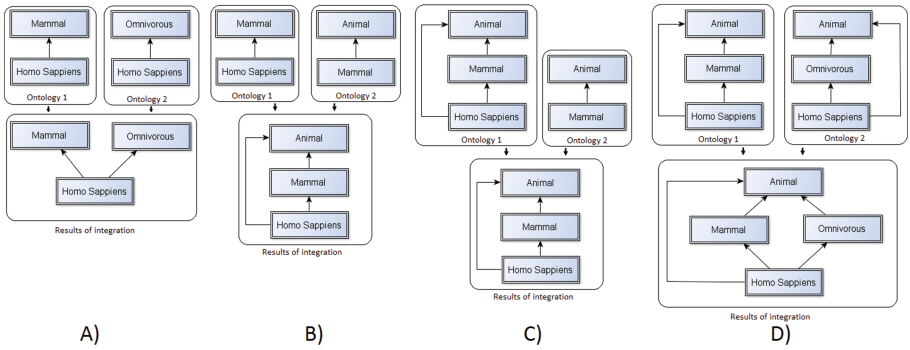

Fig. 1. Examples of ontologies integration at hierarchy level

In the second case (Fig. 1B), input ontologies seem to be very similar to the previous one. As in the previous example, any hierarchy is included in the other one, therefore $\delta_H^- = 1$. However, $H* \setminus H_1 \cup H_2 = 1$, then $\delta_H^+ = \frac{1}{2}$ and $\delta_H = \frac{3}{2}$. In this situation we "create" a new knowledge during the integration. If we consider inputs separately, we only know that *a Homo Sapiens* is *a Mammal* and *a Mammal* is *an Animal*. After the integration, we additionally know that each *Homo Sapiens* is also *an Animal*, therefore, we have found out something new. This knowledge has not been included in any of input ontologies. In this point of view, the presented integration process is very beneficial.

The next case (Fig. 1C) presents a situation where the whole set of hierarchy of Ontology 2 is included within a hierarchy of Ontology 1, formally $H_2 \subseteq H_1$. Therefore, $\delta_H^- = \delta_H^+ = 0$ and the integration process neither increases nor decreases the knowledge about that instance ($\Delta_H = 0$). Eventually, we can say that the integration of these two ontologies in not beneficial from the knowledge increase point of view.

The last example (Fig. 1D) is the most complex. The hierarchies of input ontologies are not included in each other, however they are some common parts i.e. each *Homo Sapiens* is *an Animal*. In this case $\delta_H^- = 1 - \frac{1}{5} = \frac{4}{5}$ and $\delta_H^+ = 0$ because of the cardinality of $H* = H_1 \cup H_2$ (no new knowledge has been created). Eventually, we get $\Delta_H = \frac{4}{5}$.

5.2 Integration of Concepts' Relations

Figure 2 presents some use case scenarios of the integration of concepts' relations. In the first one (Fig. 2A) two input ontologies (with two different relations) are integrated. It is easy to calculate that the potential knowledge increase in this case is maximal and equal to 1.

The second case (Fig. 2B) represents the situation where relation from ontology 2 is more general than the relation in ontology 1: "*is parent*" $\leftarrow$ "*is mother*". The common part of inputs ontologies is pair: (woman, boy) and (woman, girl). The more general relation can not be replaced by more detailed because it cause

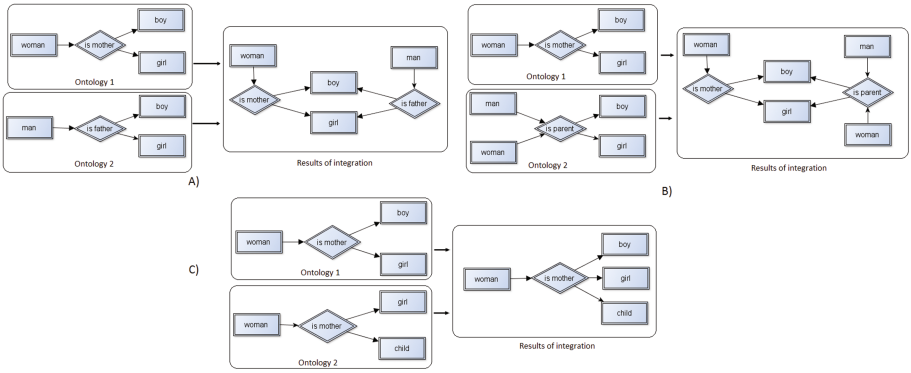

Fig. 2. Examples of ontologies integration at relation level

the lost information that man is parent of boy and man is parent of girl. Therefore $\Delta_R = \frac{3}{2}$. Due to the fact that $\omega = 2$, the final knowledge increase is equal to $\frac{3}{4}$.

In the last example (Fig. 2C) relations in inputs ontologies are equivalency. The "new knowledge" is contained only in the second ontology and it is pair *(woman, child)*. Therefore, $|r \cap r'| = 1$ and $|r \cup r'| = 3$, so $\Delta_R = \frac{2}{3}$ and $\omega = 1$. Finally, we can say that the ontology integration on the relation level increases our knowledge by $\frac{2}{3}$.

6 Future Works and Summary

This paper is a straight continuation of our previous research [5,6], which addressed the problem of ontology integration on the concept and instance levels. This article is devoted to the problem of ontology integration on the concepts' relation level. Due to the limited space we have omitted the integration of relations that exist not only on a "schema" level of concepts, but actually describe interactions of instances of concepts.

The main contribution of the paper is a set of algorithms of ontology integration on relation level. These algorithms distinguish the integration into the procedure of combining relations between concepts and the integration of concept's hierarchies (which in our opinion entail too many consequences and restrictions to be integrated using the same algorithm as relations). The article also contains a set of algorithms that can be used to evaluate the valency of the performed integration. Every procedure is carefully analysed and the outcomes are described with illustrative examples.

In the future, we plan to extend the framework of knowledge increase integration of aforementioned relations between instances. We also plan to investigate the usefulness of our ideas in other knowledge or data representation methods, such as federated data warehouses.

References

1. Bobrowski, M., Marré, M., Yankelevich, D.: Measuring data quality. Universidad de Buenos Aires. Report. 1999:99–002 (1999)
2. Flahive, A., Taniar, D., Rahayu, W.: Ontology as a Service (OaaS): a case for sub-ontology merging on the cloud. J. Supercomput. **65**, 185–216 (2013). doi:10.1007/s11227-011-0711-4
3. Frank, A.U.: Data quality ontology: an ontology for imperfect knowledge. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds.) COSIT 2007. LNCS, vol. 4736, pp. 406–420. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74788-8_25
4. Geisler, S., Weber, S., Quix, C.: An ontology-based data quality framework for data stream applications. In: 16th International Conference on Information Quality, pp. 145–159 (2011)
5. Kozierkiewicz-Hetmańska, A., Pietranik, M.: The knowledge increase estimation framework for ontology integration on the concept level. J. Intell. Fuzzy Syst. **32**(2), 1161–1172 (2017). doi:10.3233/JIFS-169116
6. Kozierkiewicz-Hetmańska, A., Pietranik, M., Hnatkowska, B.: The knowledge increase estimation framework for ontology integration on the instance level. In: Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B. (eds.) ACIIDS 2017. LNCS, vol. 10191, pp. 3–12. Springer, Cham (2017). doi:10.1007/978-3-319-54472-4_1
7. Le, D.H., Dang, V.T.: Ontology-based disease similarity network for disease gene prediction Vietnam (2016). doi:10.1007/40595-016-0063-3
8. Lozano-Tello, A., Gómez-Pérez, A.: Ontometric: a method to choose the appropriate ontology. J. Database Manage. **2**(15), 1–18 (2004)
9. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management. Springer, London (2008). doi:10.1007/978-1-84628-889-0
10. Pietranik, M., Nguyen, N.T.: A multi-atrribute based framework for ontology aligning. Neurocomputing **146**, 276–290 (2014). doi:10.1016/j.neucom.2014.03.067
11. Porello, D., Endriss, U.: Ontology merging as social choice: judgment aggregation under the open world assumption. J. Logic Comput. **24**(6), 1229–1249 (2014)
12. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS, vol. 2473, pp. 251–263. Springer, Heidelberg (2002). doi:10.1007/3-540-45810-7_24
13. Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P., Aleman-Meza, B.: OntoQA: metric-based ontology quality analysis (2005). http://lsdis.cs.uga.edu/library/download/OntoQA.pdf