A description logics formalization for the ontology matching

Manel Kolli a *, Zizette Boufaida a

a LIRE Laboratory, Computer Science Department, Mentouri University of Constantine 25000, Algeria

Abstract

In this paper, we propose a new solution to the ontology matching problem. The deduction of the relations among the semantic entities is done by aggregating or composing the relations between their subsumers, which are already deduced by using the semantic distance. The results are validated with the description logics (DLs) mechanisms. Furthermore, our approach is presented in a formal way by exploiting the mechanisms of DLs reasoning. This allows us to have complete and precise results. During the process, we use an example to well explain how the proposed approach works.

Keywords
Ontology matching; description logics; semantic relations; semantic web.

1. Introduction

Detecting semantic relations (equivalence, subsumption, disjunction, ...) among ontology entities is a big challenge especially in open environments like the semantic web, where the most applications are based on the use of ontologies for representing theirs domains. Thus, the application interoperability cannot be realized only by finding the semantic mappings between the entities belonging to these ontologies. The process of finding the semantic mappings is called ontology matching. The set of these mappings is called ontology alignment.

Although there are many ontology matching approaches in the literature but most of them are only based on the measures of similarity between the semantic entities for the ontology matching process. These measures have as result, a value that determines the similar entities (only equivalence relation) of the compared ontologies. This means that these measures are not sufficient for the ontology integration. Furthermore, they cannot solve all the equivalences problems among the semantic entities for example we can have the same value for two different calculations because they rely on syntactic and structural criteria [1]. Evaluation studies have shown that existing approaches often trade off precision and recall. The resulting mapping either contains a fair amount of errors or only covers a small part of the ontologies involved [2], [3] and [4].

In order to minimize the amount of errors, we previously have developed a new system based on a reliable tool, which was the case-based reasoning (CBR) for the detection of semantic relations among the semantic entities of ontologies [1]. The cycle of our system consists of five steps: elaboration, retrieving, reuse, revision and memorization. Notice that the "reuse" represents the key step particularly with regard to adaptation knowledge,

* Manel Kolli. Tel.: +967711978568.
E-mail address: kollimanel@yahoo.fr.
which we have suggested in form of rules. These rules are not complete and sometimes require human intervention. This means that the application of these rules is not sufficient for extracting elements of information from the data.

In the proposed system, the ontologies are expressed as Description Logics (DLs) knowledge bases. It is well known that, the semantic entities are organized according to the mechanism of subsumption in their ontologies. This organization implies the existence of these entities in the intentional definitions of their subsumee. Thus, the detection of the semantic relations among semantic entities can be done through the combination of the semantic relations of their subsumers with small modifications. This combination may be realised by the composition and the aggregation operations of the relations between their subsumers.

In this study, we aim at augmenting the effectiveness of our ontology matching system by presenting the adaptation knowledge in a formal way. This allows us to have complete and precise results by exploiting the mechanisms of the DLs reasoning. To accomplish our goal, we propose a new solution to the matching problem. We build our solution in two levels. In the first, the process of ontology matching (M1) consists of comparing all the semantic entities of two ontologies $O_1$ and $O_2$ by applying the similarity measure, which is the semantic distance inspired by [5] and adapted to the DLs in [6] in order to infer the semantic relation between them. In the second, the process of matching (M2) completes the comparison of the primitive semantic entities, which have a fuzzy result. After, it compares the defined semantic entities. The deduction of the relations among them is done by aggregating or composing the relations of their subsumers that are already deduced. The results obtained by these operations are validated with the DLs mechanisms. This approach returns a generic and extensible ontology of the semantic relations among the semantic entities (ontology of alignment). We describe this ontology with the formalism of conceptual graph that allows us to represent the cases in a form precise, readable and usable by a computer.

The rest of this paper is organized as follow: In section 2, we recall the basic definitions of DLs that we have used in this work and their classical properties. Section 3 shows more details on our ontology matching approach. Our conclusion and future work are described in the final section.

2. Basics of DLs

We start with a few definitions that characterize the DLs formalism.

2.1. Definition 1 (interpretation notion)

Let us consider $C$ and $D$ two concepts. An interpretation $I = (\Delta, \cdot_1)$ consists of a set $\Delta$ (the domain of $I$) and a function $\cdot_1$ (the interpretation function of $I$) that maps every concept to a subset of $\Delta$ such that:

- $\cdot_1^1 = \emptyset$
- $\cdot_1^{(C \cap D)} = C^I \cap D^I$
- $\cdot_1^{(C \cup D)} = C^I \cup D^I$
- $\cdot_1^{(\neg C)} = \Delta \setminus C^I$

2.2. Definition 2 (subsumption, equivalence, disjunction and overlapping)

- A concept $D$ is subsumed by a concept $C$ (respectively $C$ subsumes $D$), which is denoted by $D \subseteq C$ (respectively $C \supseteq D$) if and only if $D^I \subseteq C^I$, ($\forall I$).
- The concepts $C$ and $D$ are equivalent, which is denoted by $C \equiv D$ if and only if $C^I = D^I$, ($\forall I$).
- A concept $C$ is disjoint from a concept $D$, which is denoted by $D \perp C$ if and only if $C^I \cap D^I = \emptyset$, ($\forall I$).
- The concepts $C$ and $D$ are overlapped, which is denoted by $C \cap D$ if and only if $C^I \cap D^I \neq \emptyset$, ($\forall I$).

3. Ontology matching

As we already have mentioned before, we have previously developed a new system based on CBR mechanism for the detection of semantic relations among the semantic entities [1]. Our system consists of retrieving the source cases from the case-base (ontology of alignment) the cases that contain the subsumers of the concerned concepts. Then, it provides a solution to the target case from the solution of the selected source cases, which are adapted in order to satisfy the constraints of the posed problem, which is here the detection of the semantic relation between
two entities. The adaptation of the retrieved cases is done in the reuse step, which consists of combining the solutions of the retrieved cases and adapting them in order to find a solution to the target case.

In this step, we rely on the model of adaptation proposed in [7]. This model is based on the techniques of knowledge discovery from databases (KDD). The objective of the KDD is to obtain knowledge from data. A KDD session usually relies on two main steps: data preparation and data-mining.

Data-mining step allows extracting elements of information from the data. We realize this extraction by applying the Adaptation Knowledge (AK), which we have suggested in the previous work in form of rules. In this study, we aim at representing the adaptation knowledge in an other way by proposing a new formal approach of finding the semantic relations between concepts, which is called **ontology matching approach**.

Our approach of ontology matching is articulated in two levels. In this section, we detail it by applying it on two ontologies from the same domain to be able to represent the special cases. The Table 1 represents the two ontologies taken from [1].

Table 1. Concepts description of the two example ontologies O1 and O2

<table>
<thead>
<tr>
<th>Ontologies name</th>
<th>Ontologies description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>Man &lt; Person</td>
</tr>
<tr>
<td></td>
<td>Nationality &lt; TOP</td>
</tr>
<tr>
<td></td>
<td>Algerian &lt; Nationality</td>
</tr>
<tr>
<td></td>
<td>Algerian_citizen = person (\ni) has_nationality.algerian</td>
</tr>
<tr>
<td>O2</td>
<td>Parent = human (\ni) has_children.human</td>
</tr>
<tr>
<td></td>
<td>Man = human (\ni) has_sex.male</td>
</tr>
<tr>
<td></td>
<td>Woman = human (\ni) has_sex.female</td>
</tr>
</tbody>
</table>

3.1. Level 1

In this level, the process of ontology matching \(M_1\) consists of comparing all the primitive semantic entities of two ontologies O1 and O2 by using semantic distance. The result will be one of the relations from the set: \(\{\equiv, \perp, \equiv\}\) (Fig 1.). Intuitively, \(C_i \equiv C_j\) means that \(C_i\) is equivalent to \(C_j\). \(C_i \perp C_j\) means that \(C_i\) is disjoint from \(C_j\). Finally, \(C_i \equiv C_j\) means that the relation between \(C_i\) and \(C_j\) is fuzzy. This fuzzy relation can be one of the relations from the set: \(\{\&\leq\equiv\gt\}\). \(C_i \leq C_j\) means that \(C_i\) is more general than \(C_j\). Thus, we can say that \(C_i\) is the subsumer of \(C_j\). \(C_i \geq C_j\) means that \(C_i\) is less general than \(C_j\). Thus, \(C_i\) is the subsumee of \(C_j\). \(C_i \& C_j\) means that \(C_i\) is overlaid with \(C_j\).

If the result of the comparison is not fuzzy i.e.: it’s one of the relations from the set: \(\{\equiv, \perp\}\) then: directly, we add these cases to the case-base (ontology \(O_{A1}\)). But, if the result of the comparison is fuzzy i.e: it’s equal to \(\equiv\) then: its precision will be done in the second level.

---

![Fig. 1. Comparison results](image-url)
The result of the comparison is a gradual increase in size of the case-base, which demonstrates the need of the organization and the maintenance of the case-base throughout the system life [1]. To reply to this need, we use the ontology notion whose role is to model this kind of knowledge. This generic and extensible ontology is called ontology of alignment $O_{A1}$. As we have already mentioned, the formalism that we use for the description of this ontology is the conceptual graph. The set of the deduced relations at this level represents the ontology alignment $A_1$ (see Fig 2).

![Fig. 2. Matching process in the first level](image)

In general way, a conceptual graph is defined as a graph with two kinds of nodes [8]: the concepts and the conceptual relations,

- The concepts, which represent in our situation the semantic entities of the two ontologies $O_1$ and $O_2$
- The conceptual relations, which symbolize the semantic relation between two concepts.

The concepts are graphically represented within hooks [Concept]; the conceptual relations are represented within parentheses (Conceptual relation), with a single entering arc (e) and a single outgoing arc (s). An entering arc connects a concept to a conceptual relation, and an outgoing arc connects a conceptual relation to a concept.

In our example, the primitive concepts are: person, man, nationality and algerian of $O_1$ and human and male of $O_2$. The comparison results among these concepts are implemented as follows:

- $\text{[person]} \xrightarrow{\text{e}_1} \text{[human]}$
- $\text{[nationality]} \xrightarrow{\text{e}_2} \text{[human]}$
- $\text{[algerian]} \xrightarrow{\text{e}_3} \text{[human]}$
- $\text{[man]} \xrightarrow{\text{e}_4} \text{[male]}$
- $\text{[person]} \xrightarrow{\text{e}_5} \text{[man]}$
- $\text{[nationality]} \xrightarrow{\text{e}_6} \text{[male]}$

The first two results will be stored in the ontology of alignment $O_{A1}$, and the last (fuzzy results) will be determined in the second level.

In this example, we have the role $\text{has_nationality}$ of $O_1$ and the two roles: $\text{has_children}$ and $\text{has_sex}$ of $O_2$. All this roles are primitives. So, The comparison results among these roles are:

- $\text{[has_nationality]} \xrightarrow{\text{e}_7} \text{[has_children]}$ and $\text{[has_nationality]} \xrightarrow{\text{e}_8} \text{[has_sex]}$

### 3.2 Level 2

At this level, we first complete the comparison of the primitive semantic entities, which have a fuzzy result. After, we compare the defined semantic entities. The deduction of the relations among them is done by aggregating or composing the relations of their subsumers, which is already deduced in the first level.

Fig 3. shows the way our matching process works ($M_2$) in the second level. It takes as input the ontology of alignment $O_{A1}$ and it generates the final ontology of alignment $O_A$. The ontology alignment $A_2$ is represented by the set of the deduced relations at this level.

![Fig. 3. Matching process in the first level](image)
3.2.1. Composition:

In the Semantic Web infrastructure that rests on the DLs for the ontology construction, we note that the semantic entities are organized according to the subsumption relation, which allows the deduction of subsumption relations (\(\subseteq\)) among the concepts of the same ontology in an easy and direct way.

This observation permits us to say that the reuse of the stored case in the ontology \(O_{A1}\) may be deduced by the composition operation. Thus, if there exists a semantic relation between concept \(C_1\) of ontology \(O_1\) and concept \(C_2\) of ontology \(O_2\), and another semantic relation which is the subsumption relation between concept \(C_2\) and concept \(C_3\) of the same ontology (\(O_2\)), then it should be possible to obtain the semantic relation between the concepts \(C_1\) and \(C_3\). The possible results of this composition are shown in Table 2.

Table 2. Composition of the semantic relations

<table>
<thead>
<tr>
<th>(C_1 \cap C_2)</th>
<th>(=)</th>
<th>(\subseteq)</th>
<th>(\supseteq)</th>
<th>&amp;</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\subseteq)</td>
<td>(\subseteq)</td>
<td>(\supseteq)</td>
<td>&amp;</td>
<td>(\equiv)</td>
<td></td>
</tr>
<tr>
<td>(\supseteq)</td>
<td>(\supseteq)</td>
<td>(\subseteq)</td>
<td>&amp;</td>
<td>(\equiv)</td>
<td></td>
</tr>
</tbody>
</table>

The proof of these results is done by the interpretation notion of DLs as follows:
1. \(((C_1 \equiv C_2) \rightarrow C_1^{i} = C_2^{i}) \land (C_2 \subseteq C_3 \rightarrow C_2^{i} \subseteq C_3^{i})) \Rightarrow (C_1 \subseteq C_1^{i} \rightarrow C_1 \subseteq C_3^{i})), \(\forall I\).
2. \(((C_1 \subseteq C_2 \rightarrow C_1^{i} \subseteq C_2^{i}) \land (C_2 \subseteq C_3 \rightarrow C_2^{i} \subseteq C_3^{i})) \Rightarrow (C_1 \subseteq C_1^{i} \rightarrow C_1 \subseteq C_3^{i})), \(\forall I\).
3. \(((C_1 \land C_2 \rightarrow (C_1 \cap C_2)^{i} \neq \emptyset)) \rightarrow (C_1 \cap C_2 \subseteq C_2^{i} \subseteq C_3^{i})) \Rightarrow ((C_1 \cap C_3)^{i} \neq \emptyset \rightarrow C_1 \cap C_3^{i}), \(\forall I\).
4. \(((C_1 \equiv C_2) \rightarrow C_1^{i} \subseteq C_2^{i}) \land (C_2 \supseteq C_3 \rightarrow C_2^{i} \supseteq C_3^{i})) \Rightarrow (C_1 \supseteq C_1^{i} \rightarrow C_1 \supseteq C_3^{i})), \(\forall I\).
5. \(((C_1 \supseteq C_2 \rightarrow C_1^{i} \supseteq C_2^{i}) \land (C_2 \supseteq C_3 \rightarrow C_2^{i} \supseteq C_3^{i})) \Rightarrow (C_1 \supseteq C_1^{i} \rightarrow C_1 \supseteq C_3^{i})), \(\forall I\).

Ever, according to the interpretation notion of DLs, we deduce that there are fuzzy results (\(\equiv\)) in the Table 2. This means that the composition of these relations generates more than one result. We can take for example the case where we have: \((C_1 \supseteq C_2)\) and \((C_2 \subseteq C_3)\). This implies that \((C_1^{i} \supseteq C_2^{i})\) and \((C_2^{i} \subseteq C_3^{i})\). So, we can deduce that: \((C_1^{i} = C_2^{i})\) or \((C_1^{i} \supseteq C_2^{i})\) or \((C_1^{i} \subseteq C_2^{i})\). In this case, the relation between \(C_1\) and \(C_3\) may be one of the relations from the set: \(\{\equiv, \supseteq, \subseteq\}\).

Thus, the application of the composition operation for these cases cannot solves the problem of relations’ deduction. For the purpose of detecting all the possible relations between all concepts, we propose to use in this level also another operation, which is the similarity aggregation.

In the preceding example, we have the case: \((\text{human} \equiv \text{person})\) stored in the ontology \(O_{A1}\). We can also deduct the two cases: \((\text{man} \supseteq \text{person})\) and \((\text{Algerian\_citizen} \supseteq \text{person})\) from the ontology \(O_1\). According to the Table 2, the application of the composition operation for these cases gives us the two resulting cases: \((\text{human} \supseteq \text{man})\) and \((\text{human} \supseteq \text{Algerian\_citizen})\).

Also, We can deduce according to the subsumption relation the following cases: \((\text{man} \supseteq \text{human})\), \((\text{woman} \supseteq \text{human})\) and \((\text{parent} \supseteq \text{human})\) from the ontology \(O_2\). According to the Table 2, the application of the composition operation for these cases gives us the resulting cases: \((\text{person} \supseteq \text{man})\), \((\text{person} \supseteq \text{woman})\) and \((\text{person} \supseteq \text{parent})\).

The memerization of these cases in the ontology \(O_A\) is implemented as follows:

Example: \[
\begin{align*}
\text{Algerian\_citizen} & \xrightarrow{v_{10}} \text{[person]} \\
\text{[man]} & \xrightarrow{v_{10}} \text{[person]} \\
\text{[woman]} & \xrightarrow{v_{12}} \text{[person]} \\
\end{align*}
\]

3.2.2. Similarity aggregation:

In general, there may be several subsumers for the same semantic entity (for this operation, we deal only with the defined entity). Thus, there are several source cases for each candidate pair of entities. Their solutions have to be combined into a single solution for the target case. We call this operation: similarity aggregation which takes as
input all the similarity relation values (solutions) of source cases obtained as the result of the previous step inorder to aggregate them in only one similarity relation value by exploiting the techniques of DLs.

In proposition 1, we give some possible combinations between two concepts. We also show how to apply the aggregation function on these concepts.

**Proposition 1:** Let C and D be two concepts that are defined such as: \( C = (\cap C_1 \ldots C_n \cup C_{n+1} \ldots C_k) \) and \( D = (\cap D_1 \ldots D_U \cup \cap D_{m+1} \ldots D_j) \). Let us considered the semantic relations \( R_1, R_2, \ldots R_n \) deduced between the concepts \( C_i \) and \( D_p \). \( Aggreg(R_1, R_2, \ldots R_n) = R \) is the aggregation function applied to the semantic relations among the different concepts that define the two concepts \( C \) and \( D \). This function returns the relation \( R \) between the two concepts \( C \) and \( D \).

The relation \( R \) can be deduced as follows:

1. If \( (\exists R_i \in \{R_1, R_2, \ldots R_n\} / s = 1 \ldots k, such as: R_i = \top) \) and (if the concepts \( C \) and \( D \) are only a conjunctions)

\[ Aggreg(R_1, R_2, \ldots R_n) = \top i.e. C \sqsubseteq D. \]

2. If \( (\forall R_i \in \{R_1, R_2, \ldots R_n\} / s = 1 \ldots k, such as: R_i = \top) \) then: \( Aggreg(R_1, R_2, \ldots R_n) = \top i.e. C \sqsubseteq D. \)

3. If \( (\forall R_i \in \{R_1, R_2, \ldots R_n\} / s = 1 \ldots k, such as: R_i = \top) \) and \( (\exists R_i / t=1 \ldots k \text{ and } t \neq s \text{ such as: } R_i = \top) \) then:

\[ Aggreg(R_1, R_2, \ldots R_n) \subseteq \top i.e. C \subseteq D. \]

4. If \( (\forall R_i \in \{R_1, R_2, \ldots R_n\} / s = 1 \ldots k, such as: R_i = \top) \) and \( (\exists R_i / t=1 \ldots k \text{ and } t \neq s \text{ such as: } R_i = \top) \) then:

\[ Aggreg(R_1, R_2, \ldots R_n) = \top \text{ i.e. } C \sqsubseteq D. \]

5. If \( (\forall R_i \in \{R_1, R_2, \ldots R_n\} / s = 1 \ldots k, such as: R_i = \& \text{ or } R_i = \equiv \text{ or } R_i = \sqsubseteq \text{ or } R_i = \subseteq) \) then: \( Aggreg(R_1, R_2, \ldots R_n) = \& \text{ i.e. } C \sqcap D. \)

It is easy to proof the proposition 1 by using the interpretation notion of DLs: Let us considered the semantic relations \( R_1, R_2, \ldots R_n \) deduced between the concepts \( C_i \) and \( D_p \).

1. \( \exists R_i \in \{R_1, R_2, \ldots R_n\} / s=1 \ldots k, such as: R_i = \top \Rightarrow \exists C_V / v = 1 \ldots i \text{ and } \exists D_a / u = 1 \ldots j, such as: C_V \sqsubseteq D_a \)

\[ \exists C_V \sqcap D_a = \emptyset (\forall i) \]

\[ \Rightarrow (\forall C_V / v = 1 \ldots i \text{ and } \exists D_a / u = 1 \ldots j, C_V \equiv D_a) \]

2. \( \forall R_i \in \{R_1, R_2, \ldots R_n\} / s=1 \ldots k, R_i = \top \Rightarrow \forall C_V / v = 1 \ldots i \text{ and } \forall D_a / u = 1 \ldots j, C_V = D_a \)

\[ \Rightarrow C_V \sqsubseteq D_a \]

3. \( (\forall R_i / t=1 \ldots k \text{ and } t \neq s, such as: R_i = \subseteq) \Rightarrow (\exists C_V / v = 1 \ldots i \text{ and } \exists D_a / u = 1 \ldots j, C_V \subseteq D_a) \)

\[ (1) \text{ and } (2) \Rightarrow C_V \sqsubseteq D_a \text{ and } C_V \sqsubseteq D_a \]

4. In the same way, we can proof the forth result by inverting the subsumption relation.

5. \( \exists R_i \in \{R_1, R_2, \ldots R_n\} / s=1 \ldots k, such as: R_i = \equiv \Rightarrow \exists C_V / v = 1 \ldots i \text{ and } \exists D_a / u = 1 \ldots j, such as: C_V \equiv D_a \)

\[ \Rightarrow C_V \equiv D_a \text{ and } C_V \equiv D_a \]

In the same way, we can proof the other combinations (\( R_i = \equiv \) or \( R_i = \sqsubseteq \) or \( R_i = \subseteq \)).

In the previous example, the defined concept of the ontology \( O_1 \): the concept Algerian_citizen and the defined concepts of the ontology \( O_2 \) are: parent, man, woman and female. Here, we begin with the comparison between the two concepts: Algerian_citizen and parent. In the case-base (ontology of alignment \( O_2 \)), we have: the concepts person and human are similar; the concepts Algerian and human are different and the roles has_nationality and has_children are different too. Indeed, the concepts Algerian_citizen and parent are overlapped (according to the rule 5). The memorization of this case in the ontology \( O_2 \) is implemented as follows:

\[[Algerian\_citizen] \xrightarrow{e_{ii}} (\&) \xrightarrow{s_{ii}} [parent]\]
After the memorization of the case ((Algerian _citizen, parent)), we note that the application of the rule 5 is also possible on the concepts: (Algerian_citizen, man) and (Algerian_citizen, woman) i.e: the concept person is the subsumer of the concept Algerian_citizen and the concept human is the subsumer of the concepts woman and man. The concepts: person and human are similar. Thus, the concepts Algerian_citizen and man are overlapped and the concepts Algerian_citizen and man are overlapped too. The memorization of the two cases in the ontology $O_A$ is implemented as follows:

$$[\text{Algerian_citizen}] \xrightarrow{e_{15}} (\&) \xrightarrow{s_{15}} [\text{woman}] \quad \text{and} \quad [\text{Algerian_citizen}] \xrightarrow{e_{16}} (\&) \xrightarrow{s_{16}} [\text{man}]$$

In the same way, we infer the semantic relations among the other different concepts of the two ontologies $O_1$ and $O_2$.

4. Conclusion

In this article we proposed a new formal approach for the ontology matching problem. This approach is based on the DLs techniques and semantic distance for the detection of the semantic relations among the semantic entities of two different ontologies. Here, the deduction of these relations is done by aggregating or composing the relations of their subsumers, which is already deduced before. The results obtained by these operations are validated by the DLs mechanisms.

This paper is a theoretical one. We have no experimental setting to demonstrate the superiority of the approach over an eventual previous one. However, we claim that it conveniently demonstrate the benefits brought by the use of DLs mechanisms within ontology matching. Furthermore, we have illustrated our approach using an example. Our future investigation aims certainly at implementing all the proposed system in OWL API language [9].

References