A Semantic Similarity Measure for Ontology-Based Information

Heiner Stuckenschmidt

University of Mannheim

Abstract. Computing the similarity between data elements is a basic functionality in flexible query answering systems. In the case of complex data definitions, for instance in terms of an ontology, computing the similarity between data elements becomes a non-trivial problem. In this paper, we propose a similarity measure for data described in terms of the DL-lite ontology language. In this measure, we take implicit information contained in the definition of classes and relations into account. In contrast to many other proposals for similarity measures, our proposal does not rely on structural criteria of the definitions involved but is solely based on the logical consequences that can be drawn.

1 Introduction

Similarity plays a central role in information management, especially in the context of environments like the semantic web where data may originate from different sources and has to be combined and integrated in a flexible way. Being able to compute meaningful measures of similarity between data and data models helps to perform a number of central tasks such as retrieval of structured data based on partial specification, data and schema integration based on the similarity of definitions in different sources, and similarity based query answering on the integrated model.

Defining similarity measures for semantic web data has to take into account that this data is often described based on ontological knowledge that defines implicit information about the data. In order to define a meaningful similarity measure for such data the similarity measure also has to consider this implicit information. In our work, we investigate such a semantic similarity measure for data that has been described using DL-lite, an efficient subset of the web ontology language OWL. In defining the similarity measure we benefit from the fact that DL-lite allows to compile implicit knowledge into explicit data statements. This allows us to first use logical reasoning for compiling the implicit data into explicit statements and then determining the similarity of the models based on the explicit statements.

More specifically, in this paper, we define a semantic similarity measure for DL-lite knowledge bases. This similarity measure provides a theoretical basis for different kinds of more practical problems such as the computation of similarity between concept definitions or similarity-based query answering.

The paper is organized as follows. We first provide a brief introduction to the DL-lite language and introduce an example ontology that is used throughout the paper, we then define a similarity measure on data models without considering the ontology and stepwise extend this definition to include positive and negative implications from the ontology into the measure. We conclude with a brief discussion of the formal properties of the measure in terms of axiomatization and computational complexity.

2 DL-Lite

The DL-lite family of logics [2] has been developed as an alternative language for encoding ontologies for data management that is better suited for large data sets than traditional description logics such as the logics $\mathcal{SHIF}$ and $\mathcal{SHOIQ}$ that form the basis for the language standards OWL-lite and OWL-DL. DL-lite restricts the use of logical operators in axioms in such a way that the resulting theories can be verified and queried efficiently using techniques known from the database area. In the following, we briefly introduce the different dialects of the DL-lite language and define a small ontology we will use as a running example throughout the paper.

Syntax We begin with defining the language of the DL-lite family. These languages define four kinds of constructs: A *basic concept* B is either a *concept name* A or an expression of the form $\exists R$ where R is a basic role as defined below. A general Concept C is either a basic concept as defined above or the negation of a basic concept denoted as $\neg B$. A *basic role* R is either a role name P or the inverse of a role name denoted as P^{-}. A *general role* E is either a basic role as defined above or the negation of a basic role denoted as $\neg R$. Based on these constructs the different languages allow the construction of different kinds of axioms that are defined in the following: All languages allow *inclusion axioms* of the form $B \sqsubseteq C$ where B is a basic concept and C is a general concept. Note that the languages also allows to have disjunctions of basic concepts on the left-hand and conjunctions of general concepts on the right hand side of the inclusion. This, however, does not extend the expressiveness of the language as the same effect can be achieved by using multiple inclusion statements without conjunction and disjunction.

All languages allow the definition of *membership assertions* that are either of the form $A(a)$ where A is a concept name and a is an object name or $P(a, b)$ where P is a role name and a and b are object names. The Language DL-lite(R) further allows *role inclusion axioms* of the form $R \sqsubseteq E$ where R is a basic role and E is a general role. The language DL-lite(F) allows to state that a basic role is *functional* denoted as $(funct\, R)$. A DL-lite knowledge base $KB = (\mathcal{T}, \mathcal{A})$ consists of a T-Box $\mathcal{T}$ and an A-box $\mathcal{A}$. The T-box contains a finite number of inclusion axioms and depending on the language might also contain Role inclusion or functional role definitions. The A-box contains a finite number of membership assertions that use concept and role names occurring in the respective T-Box.

Examples In order to illustrate the ability of DL-lite to represent simple ontologies and conceptual models, we use a small example from the teaching domain. We start describing $Professor$s and $Student$s in terms of the relation between them. We define that a $Professor$ always works somewhere as the title of a professor is bound to a certain position, further, we say that $Professor$s are always mentoring someone and that the

$hasMentor$-Relation holds between $Student$s and $Professor$s The corresponding DL-lite T-Box is the following:

$$Professor \sqsubseteq Person \tag{1}$$
$$Student \sqsubseteq Person \tag{2}$$
$$Professor \sqsubseteq \exists worksAt \tag{3}$$
$$Professor \sqsubseteq \exists hasMentor^{-} \tag{4}$$
$$\exists hasMentor \sqsubseteq Student \tag{5}$$
$$\exists hasMentor^{-} \sqsubseteq Professor \tag{6}$$

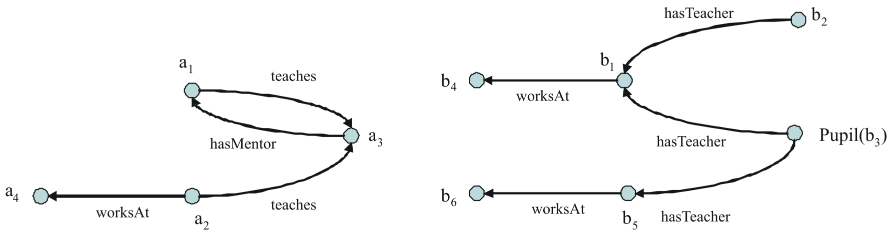
We continue with describing $Teacher$s and $Pupil$s by stating that $Pupil$s are always in the $hasTeacher$ relation and that this relation always holds between a $Person$ and a $Teacher$. We also say that the $hasTeacher$ relation has an inverse relation $Teaches$. Finally, we say that $Pupil$s and $Student$s are disjoint classes as everyone has to finish school before becoming a $Student$. The corresponding DL-lite T-Box is the following:

$$Teacher \sqsubseteq Person \tag{7}$$
$$Pupil \sqsubseteq Person \tag{8}$$
$$Pupil \sqsubseteq \exists HasTeacher \tag{9}$$
$$\exists hasTeacher^{-} \sqsubseteq Teacher \tag{10}$$
$$\exists hasTeacher \sqsubseteq Person \tag{11}$$
$$hasTeacher^{-} \sqsubseteq Teaches \tag{12}$$
$$Student \sqsubseteq \neg Pupil \tag{13}$$

We will use this situation as a running example for illustrating our approach for defining similarity on a semantic level. For a formal definition of the semantics of DL-lite, we refer to [2]

3 A-Box Similarity

We start our investigation of similarity in DL-lite by looking only at the A-Box. As mentioned in section 2 an A-box is a set of membership statements of the following form: $C(a), P(a, b)$ where a,b are constants, C is a concept name and P is a property name. Further, we extend the notion of an A-Box by also allowing membership statements of the form $\neg C(a)$ and $\neg P(a, b)$ stating that object a is not a member of Concept C and that the objects a and b are not in relation R, respectively. Note that these additional statements do not change the logic used. The statement $\neg C(a)$ can be represented by a combination of the statements: $D(a)$ and $D \sqsubseteq \neg C$ while $\neg P(a, b)$ can be represented in terms of $Q(a, b)$ and $Q \sqsubseteq \neg P$. For convenience, we denote the set of constants in an A-box as $const(\mathcal{A})$. In in the following we define a basic similarity measure for comparing A-boxes extended in the way described that will later be used as a basis for measuring similarity between concepts and complete knowledge bases. We introduce the notion of A-Box similarity using the following example A-Boxes and their corresponding graphs:



(a) Graph for A-Box $\mathcal{A}_1$

(b) Graph for A-Box $\mathcal{A}_2$

Fig. 1. A-boxes used as running example in the paper.

Example 1.

$$\begin{aligned}\mathcal{A}_1 = \{&hasMentor(a_3, a_1), worksAt(a_2, a_4), \\ &teaches(a_1, a_3), teaches(a_2, a_3), \} \end{aligned} \tag{14}$$

$$\begin{aligned}\mathcal{A}_2 = \{&hasTeacher(b_2, b_1), hasTeacher(b_3, b_1), hasTeacher(b_3, b_5), \\ &worksAt(b_1, b_4), worksAt(b_5, b_6), Pupil(b_3)\} \end{aligned} \tag{15}$$

A-Boxes can be regarded as labeled directed multi-graphs, where object constants are represented by nodes and binary relations between objects are represented by links labeled with the name of the corresponding relation. This means that we can use existing graph similarity measures for measuring the similarity of the graphs representing the A-boxes. In particular, we adopt the similarity measure for labeled graphs proposed in [3] and apply it to the case of A-Boxes as described below.

The basis for determining similarity of labeled graphs is a mapping between nodes in the graphs to be compared. This mapping indicates possibly equivalent objects in both structures. We adapt this idea of a mapping between nodes to the case of A-boxes by defining a mapping to connect object constants that potentially represent the same individual in a domain. This excludes object constants of incompatible types. In our example A-boxes, for instance, the constant a_1 can be mapped on b_1 in the other models, but if we take the definitions in the T-Box into account a_2 should not be mapped on b_3 because we defined that students cannot be pupils at the same time.

For the moment however, we do not take T-Box definitions into account and formalize similarity of A-Boxes purely on the basis of explicit membership statements starting with the notion of an A-Box mapping.

Definition 1 (A-Box Mapping). *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be A-boxes, then an A-box mapping of $\mathcal{A}_1$ and $\mathcal{A}_2$ is a one-to-one relation $m : const(\mathcal{A}_1) \times const(\mathcal{A}_2)$ such that none of the following holds for any $(a_1, b_1), (a_2, b_2) \in m$:*

- $C(a_1) \in \mathcal{A}_1$ *and* $\neg C(b_1) \in \mathcal{A}_2$
- $\neg C(a_1) \in \mathcal{A}_1$ *and* $C(b_1) \in \mathcal{A}_2$
- $P(a_1, a_2) \in \mathcal{A}_1$ *and* $\neg P(b_1, b_2) \in \mathcal{A}_2$
- $\neg P(a_1, a_2) \in \mathcal{A}_1$ *and* $P(b_1, b_2) \in \mathcal{A}_2$

This limitation of the mapping limits our ability of freely map objects onto each other. Intuitively, mapping one object onto another can be seen as a hypothesis that they are the same object. Of course, this only makes sense for objects that are not in disjoint classes. In our example this means that we can map an object of type Professor onto an object of type Teacher, but not an object of type Student onto an object of type Pupil. We will discuss the impact of this restriction on the similarity of objects later.

Based on the notion of an A-Box mapping, we can define a degree of overlap between the two A-Boxes by counting the number of statements they have in common when assuming that mapped object constants represent the same real world object. For this purpose, we count the number of objects mapped onto each other that belong to the same concept in both A-Boxes as well as the pairs of objects mapped onto each other that are in the same relation to each other. Formally, the overlap of two A-Boxes is defined as follows:

Definition 2 (A-Box Overlap). *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be A-boxes, m an A-Box mapping for $\mathcal{A}_1$ and $\mathcal{A}_2$ then the overlap between $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to m is defined as:*

$$\begin{aligned}overlap(\mathcal{A}_1, \mathcal{A}_2, m) =_{def} & \ |\{C(a) | C(a) \in \mathcal{A}_i \wedge C(m(a)) \in \mathcal{A}_j, i \neq j \in \{1, 2\}\} \cup \\ & \{R(a, b) | R(a, b) \in \mathcal{A}_i \wedge R(m(a), m(b)) \in \mathcal{A}_j, i \neq j \in \{1, 2\}\}|\end{aligned}$$

Based on the amount of overlap, we can now proceed to define the similarity of two A-boxes in terms of the relative overlap of the models compared to the maximal possible overlap in the case of identical models. Formally, the similarity is defined as follows:

Definition 3 (A-Box Similarity). *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be A-boxes and m is an A-Box mapping for $\mathcal{A}_1$ and $\mathcal{A}_2$, then the similarity of $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as:*

$$Sim(\mathcal{A}_1, \mathcal{A}_2) =_{def} max_m \frac{overlap(\mathcal{A}_1, \mathcal{A}_2, m)}{|\mathcal{A}_1 \uplus \mathcal{A}_2|}$$

where $\mathcal{A}_1 \uplus \mathcal{A}_2$ denotes the disjoint union of role and concept membership statements in $\mathcal{A}_1$ and $\mathcal{A}_2$.

In the following example, we illustrate the use of this similarity measure on our example A-Boxes. We also argue that this measure already carries some semantic information in terms of the condition that a mapping can only exist between compatible object (compare definition 1).

Example 2. We first have to compute the overlap between the two A-boxes. The A-Box mapping that maximizes this overlap is the one that contains $m(a_2, b_5)$ and $m(a_4, b_6)$ as this creates an overlap with respect to the worksAt relation between the mapped constants. This means that the overlap between the two models is 2. As the size of the disjoint union of the slot membership statements in both models is 10, we get the following similarity:

$$Sim(\mathcal{A}_1, \mathcal{A}_2) = \frac{2}{10} = 0.2$$

4 Similarity of DL-lite Knowledge Bases

So far, we have only considered similarity of A-Boxes without taking the definitions of the T-Box into account. A truly semantic similarity measure, however, also has to take implicit information into account that can be derived from the T-Box [5]. In this section, we show that the measure introduced above can be extended to take these definitions into account. Different from existing approaches for computing similarity of description logic expressions, we do not specify the similarity for T-Box operators, we rather extend the A-Box of a knowledge base with implicit information. This extension uses the concept of a chase [8] that has also been identified as a suitable basis for testing the satisfiability of DL-lite Knowledge bases. We first investigate the inclusion of positive information into the A-box based on the chase algorithms described in [2]. We then extend this algorithm to also support the inclusion of negative information into the model. This second part is crucial as the inclusion of negative information has an impact on the possible A-box mappings and therefore carries semantic information by excluding mappings between incompatible objects from the overlap of the A-Boxes involved.

4.1 Positive inclusion statements

Positive inclusion statements are inclusion statements that do not have a negation on the right-hand side of the statement. The effect of these kinds of statements can completely be determined in terms of derivable A-box statements. Given a Knowledge base $KB = (\mathcal{T}, \mathcal{A})$, such derivable statements can be computed by iterative applying the expansion rules from table 1.

Applying these rules until new information is generated results in a new A-Box $pos_{\mathcal{T}}(\mathcal{A})$ that explicitly contains all implicit positive information encoded in the T-Box of KB.

Example 3. Applying the derivation rules described in table 1 to the A-Boxes from example 1 with respect to the definitions in the T-Box described in section 2 leads to the following extended A-boxes:

$$\begin{aligned} pos_{\mathcal{T}}(\mathcal{A}_1) = \{ & hasMentor(a_3, a_1), worksAt(a_1, a_5), teaches(a_1, a_3), \\ & teaches(a_2, a_3), worksAt(a_2, a_4), Person(a_1), Person(a_3), \\ & Professor(a_1), Student(a_3), \} \end{aligned}$$

if	$\mathcal{T}$ contains	$\mathcal{A}$ contains	then	add to $\mathcal{A}$	unless
	$A_1 \sqsubseteq A_2$	$A_1(a)$		$A_2(a)$	-
	$A \sqsubseteq \exists P$	$A(a)$		$P(a, b)$	$\exists b' : P(a, b') \in \mathcal{A}$
	$A \sqsubseteq \exists P^-$	$A(a)$		$P(b, a)$	$\exists b' : P(b', a) \in \mathcal{A}$
	$\exists P \sqsubseteq A$	$P(a, b)$		$A(a)$	-
	$\exists P^- \sqsubseteq A$	$P(a, b)$		$A(b)$	-
	$\exists P_1 \sqsubseteq \exists P_2$	$P_1(a, b)$		$P_2(a, c)$	$\exists c' : P_2(a, c') \in \mathcal{A}$
	$\exists P_1^- \sqsubseteq \exists P_2^-$	$P_1(a, b)$		$P_2(c, b)$	$\exists c' : P_2(c', b) \in \mathcal{A}$
	$\exists P_1 \sqsubseteq \exists P_2^-$	$P_1(a, b)$		$P_2(c, a)$	$\exists c' : P_2(c', a) \in \mathcal{A}$
	$\exists P_1^- \sqsubseteq \exists P_2$	$P_1(a, b)$		$P_2(b, c)$	$\exists c' : P_2(b, c') \in \mathcal{A}$
	$P_1 \sqsubseteq P_2$	$P_1(a, b)$		$P_2(a, b)$	-
	$P_1^- \sqsubseteq P_2^-$	$P_1(a, b)$		$P_2(a, b)$	-
	$P_1^- \sqsubseteq P_2$	$P_1(a, b)$		$P_2(b, a)$	-
	$P_1 \sqsubseteq P_2^-$	$P_1(a, b)$		$P_2(b, a)$	-

Table 1. Derivation Rules for positive information

$$\begin{aligned} pos_{\mathcal{T}}(\mathcal{A}_2) = \{ & hasTeacher(b_3, b_1), hasTeacher(b_3, b_5), hasTeacher(b_2, b_1), \\ & teaches(b_1, b_2), teaches(b_1, b_3), teaches(b_5, b_3), worksAt(b_5, b_6), \\ & worksAt(b_1, b_4), Person(b_1), Person(b_2), Person(b_3), \\ & Person(b_5), Pupil(b_3), Teacher(b_5), Teacher(b_1), \} \end{aligned} \tag{16}$$

The extended A-boxes contain additional information about the types of object constants as well as some new role membership statements, in particular $Teaches(b_1, b_2)$, $Teaches(b_1, b_3)$ and $Teaches(b_5, b_3)$ in $pos_{\mathcal{T}}(\mathcal{A}_2)$ which are drawn as dashed lines in the figure as these new statements have a direct impact on the similarity of the two A-Boxes.

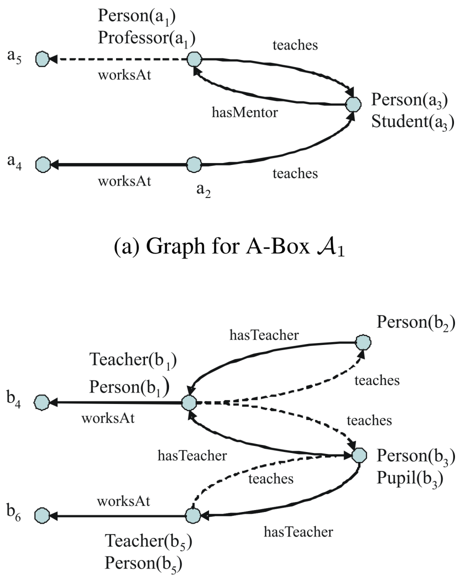
We can use these expansion rules for taking into account positive T-Box information when computing the similarity of two A-Boxes $\mathcal{A}_1$ and $\mathcal{A}_2$ by first computing the extended A-Boxes $pos_{\mathcal{T}}(\mathcal{A}_1)$ and $pos_{\mathcal{T}}(\mathcal{A}_2)$ and then computing the similarity of these extended A-Boxes rather than the similarity of the original ones. This gives us a new similarity measure for A-boxes that is closer to our idea of a semantic similarity measure than the previous definitions.

Definition 4 (Positive Semantic Similarity). *Given a DL-lite T-Box $\mathcal{T}$ and two DL-lite A-boxes $\mathcal{A}_1$ and $\mathcal{A}_2$, the positive semantic similarity of $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to $\mathcal{T}$ is defined as follows:*

$$posSim_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) =_{def} Sim(pos_{\mathcal{T}}(\mathcal{A}_1), pos_{\mathcal{T}}(\mathcal{A}_2))$$

This new definition of similarity better captures the actual similarity between the models as it takes statements into account that logically follow from the explicit statements and the definitions in the T-Box. We illustrate the impact on the similarity of knowledge bases using our running example.

Example 4. In our running example, the application of the extension rules significantly increases the overlap between the A-Boxes. Now the mapping with the maximal overlap is the one that maps a_3 on b_3, a_1 on b_1, a_2 on b_5 as well as a_5 on b_4 and a_4 on b_6. Using this mapping, the overlap between the two models contain not less than four



(a) Graph for A-Box $\mathcal{A}_1$

(b) Graph for A-Box $\mathcal{A}_2$

Fig. 2. A-boxes after adding derivable statements

elements, namely the two worksAt and two Teaches relations. As the overall number of role inclusion statements in the disjoint union has increased as well and contains statements 22, the new similarity is computed as follows:

$$posSim_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) = \frac{12}{24} = 0.5$$

Note, that so far we have not made any use of the inferred concept membership assertions as these are only relevant when negative information is available.

4.2 Negative inclusion statements

As we have seen, the inclusion of derivable information leads to a significant increase of the similarity of the two knowledge bases. The high similarity can partially be explained by the fact that our definition is very liberal about matching constants onto each other. In the definition 1 we claim that only compatible constants should be mapped onto each other. So far, this restriction only applies in cases, where negated membership statements are explicitly included in the A-box. In practice,negative information is seldom directly included in an A-Box but is normally derived from negative inclusion statements in the T-Box. In order for such implied negated statements to have an impact on the similarity estimation, we need to make them explicit in the same way, we did for implied positive statements. This can be done in a two-step process. In the first step, we derive all implied negative inclusion statements that follow from the T-Box. In a second step, we use these inclusion statements to extend the A-Box with negated membership statements. The first step consists of computing the closure $cln(\mathcal{T})$ of negative inclusion statements. This is done by initializing $cln(\mathcal{T})$ with the set of inclusion statements that contain negation on the left hand side and successively applying the following set of rules until no new statements can be derived.

if	$\mathcal{T}$ contains	and $cln(\mathcal{T})$ contains	then	add to $cln(\mathcal{T})$
	$B_1 \sqsubseteq B_2$	$B_2 \sqsubseteq \neg B_3$		$B_1 \sqsubseteq \neg B_3$
	$B_1 \sqsubseteq B_2$	$B_3 \sqsubseteq \neg B_2$		$B_1 \sqsubseteq \neg B_3$
	$R_1 \sqsubseteq R_2$	$\exists R_2 \sqsubseteq \neg B$		$\exists R_1 \sqsubseteq \neg B$
	$R_1 \sqsubseteq R_2$	$B \sqsubseteq \neg \exists R_2$		$\exists R_1 \sqsubseteq \neg B$
	$R_1 \sqsubseteq R_2$	$\exists R_2^- \sqsubseteq \neg B$		$\exists R_1^- \sqsubseteq \neg B$
	$R_1 \sqsubseteq R_2$	$B \sqsubseteq \neg \exists R_2^-$		$\exists R_1^- \sqsubseteq \neg B$
	$R_1 \sqsubseteq R_2$	$R_2 \sqsubseteq \neg R_3$		$R_1 \sqsubseteq \neg R_3$
	$R_1 \sqsubseteq R_2$	$R_3 \sqsubseteq \neg R_2$		$R_1 \sqsubseteq \neg R_3$
	-	$\exists R \sqsubseteq \neg \exists R$		$\exists R^- \sqsubseteq \neg \exists R^-$
	-	$\exists R^- \sqsubseteq \neg \exists R^-$		$\exists R \sqsubseteq \neg \exists R$
	-	$\exists R \sqsubseteq \neg \exists R$		$R \sqsubseteq \neg R$
	-	$\exists R^- \sqsubseteq \neg \exists R^-$		$R \sqsubseteq \neg R$
	-	$R \sqsubseteq \neg R$		$\exists R^- \sqsubseteq \neg \exists R^-$
	-	$R \sqsubseteq \neg R$		$\exists R \sqsubseteq \neg \exists R$

Table 2. Completion Rules for negative inclusion statements

Based on this notion of a closure of negative inclusion statements, we can now also define additional rules for extending an A-Box with derivable negative statements.

if	$cln(\mathcal{T})$ contains	$pos_{\mathcal{T}}(\mathcal{A})$ contains	then	add to $pos_{\mathcal{T}}(\mathcal{A})$	for all
	$A \sqsubseteq \neg B$	$A(a)$		$\neg B(a)$	-
	$A \sqsubseteq \neg \exists P$	$A(a)$		$\neg P(a, x)$	$x \in const(\mathcal{A})$
	$A \sqsubseteq \neg \exists P^-$	$A(a)$		$\neg P(x, a)$	$x \in const(\mathcal{A})$
	$\exists P \sqsubseteq \neg B$	$P(a, b)$		$\neg B(a)$	-
	$\exists P_1 \sqsubseteq \neg \exists P_2$	$P_1(a, b)$		$\neg P_2(a, x)$	$x \in const(\mathcal{A})$
	$\exists P_1 \sqsubseteq \neg \exists P_2^-$	$P_1(a, b)$		$\neg P_2(x, a)$	$x \in const(\mathcal{A})$
	$\exists P^- \sqsubseteq \neg B$	$P(a, b)$		$\neg B(a)$	-
	$\exists P_1^- \sqsubseteq \neg \exists P_2$	$P_1(a, b)$		$\neg P_2(b, x)$	$x \in const(\mathcal{A})$
	$\exists P_1^- \sqsubseteq \neg \exists P_2^-$	$P_1(a, b)$		$\neg P_2(x, b)$	$x \in const(\mathcal{A})$
	$P_1 \sqsubseteq \neg P_2$	$P_1(a, b)$		$\neg P_2(a, b)$	-
	$P_1 \sqsubseteq \neg P_2^-$	$P_1(a, b)$		$\neg P_2(b, a)$	-
	$P_1^- \sqsubseteq \neg P_2$	$P_1(a, b)$		$\neg P_2(b, a)$	-
	$P_1^- \sqsubseteq \neg P_2^-$	$P_1(a, b)$		$\neg P_2(a, b)$	-

Table 3. Derivation rules for negative membership statements

Applying these rules until new information is generated results in a new A-Box $ext_{\mathcal{T}}(\mathcal{A})$ that explicitly contains all implicit positive and negative information encoded in the T-Box of KB. This extended A-Box again provides us with a new version of the similarity measure. In order to also take implicit negative information into account, we define a third notion of similarity that is based on the overlap between the extended A-Boxes.

Definition 5 (Semantic Similarity). *Given a DL-lite T-Box $\mathcal{T}$ and two DL-lite A-boxes $\mathcal{A}_1$ and $\mathcal{A}_2$, the extended semantic similarity of $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to $\mathcal{T}$ is defined as follows:*

$$extSim_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) =_{def} Sim(ext_{\mathcal{T}}(\mathcal{A}_1), ext_{\mathcal{T}}(\mathcal{A}_2))$$

We illustrate the use of this extended notion of semantic similarity using our running example.

Example 5. In our example T-Box, we only have a single negative inclusion statement, namely $student \sqsubseteq \neg pupil$. There are no interactions with other axioms in the T-Box, therefore closure $cln(\mathcal{T})$ only contains this single axiom. Applying the derivation rules for negative membership statements from table 2 generates one new negative statement in A-Box $\mathcal{A}_1$, namely $\neg Student(b_3)$. This means, that we cannot map a_3 to b_3 any more according to definition 1. The possible mapping that maximizes the overlap is now the one that maps a_3 on b_2 instead. The corresponding similarity induced by this mapping is

$$extSim_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) = \frac{10}{25} = 0.4$$

This similarity makes use of all the semantic information contained in the T-Box and therefore supposedly is the most accurate estimation of the real similarity of the models compared.

5 Formal Properties of the Measure

In this section, we take a brief look at the formal properties of the similarity measure defined in this paper. In particular, we look at some axiomatic properties of the measure and investigate the complexity of computing the measure. Taking a careful look at the complexity is very important in the context of semantic similarity as taking ontological knowledge into account can quickly lead to intractability.

5.1 Axiomatic Properties

Measures of similarity are normally assumed to have some axiomatic properties. In particular, for a similarity measure sim, it is normally assumed that the following axioms hold:

1. $0 \leq sim(x, y) \leq 1$
2. $Sim(x, x) = 1$
3. $Sim(x, y) = Sim(y, x)$
4. $Sim(x, y) + Sim(y, z) \geq Sim(x, z)$

Testing our measure against these properties reveals that properties 1-3 hold for our measure, whereas property 4 does not hold. Property 1 immediately follows from the definition of measure. The overlap between two A-Boxes is at least 0 if there is no overlap and two times the size of the smaller A-Box. In case that the two A-Boxes have the same size it can happen that it has the same size as the disjoint union of the two A-Boxes which leads to a maximal similarity of 1. In particular, this is the case, if the two A-boxes are identical which establishes property 2. Property 3 also immediately follows from the definition. As m is a one-to-one relation, a mapping that holds in one direction can also be established in the other direction which means that the similarity in the opposite direction is at least as high as the other one. If it would be higher, it means that there is an additional pair of objects in m that increases the overlap, this however means that this pair could also be added to the original direction also increasing the similarity here. We can easily construct a counter-example for property 4 by choosing the A-boxes such that $\mathcal{A}_1 = \mathcal{A}_2$ and $\mathcal{A}_3 = \emptyset$. In this setting, we have $Sim(\mathcal{A}_1, \mathcal{A}_3) + Sim(\mathcal{A}_3, \mathcal{A}_2) = 0$ whereas $Sim(\mathcal{A}_1, \mathcal{A}_2) = 1$.

5.2 Complexity

Computing semantic similarity as defined in this paper is polynomial in the size of the knowledge base. In order to show this, we first have to investigate the basic version of the measure without taking the T-box into account: As pointed out in [3] the number of all possible mappings is exponential in the size of the A-Boxes. As we restrict ourselves to one-to-one mappings, the number of mappings we have to consider is significantly smaller, but it is still in $O((n+1)!)$ where n is the size of the larger A-box. When moving to the semantic versions of the similarity measure, additional complexity is added because the extended versions of the A-box can be significantly larger than the original one. As stated in [2], however, the size of the expanded model is polynomial in the size of the input knowledge-base. This means that computing the measure is in $O((n+1)!)$ where n is polynomial in the size of the knowledge base.

6 Related Work

A number of approaches for computing the similarity of complex concept expressions have been proposed. Borgida and other compared different principled ways of determining the similarity of concept expressions, in particular feature-based, network-based and information-content based approaches and discuss the pros and cons of these different approaches [1]. They conclude that network-based methods suffer from the need to combine different measures in a rather arbitrary way. According to the paper feature-based methods work in principle, but suffer from the problem of defining the right set of features. The authors conclude that measures based on information contents are most promising, provided that we find ways to estimate the correct distribution of instances in the real world. D'Amato and others propose to use the degree of overlap between the instances of two concepts for determining the similarity of concepts [4] which meets out intuitions of similarity.Hu and others define a similarity measure based solely on the definition of the concepts involved and do not require any instance information. The idea of the approach is to normalize concept descriptions into a sets of characteristic signatures that roughly correspond to approximations of models for the corresponding formula [6]. Based on this representation of concepts in terms of signatures, they apply standard information retrieval techniques for determining the similarity between the signatures. In this step, however, much of the information contained in the concept expression is lost as the similarity is determined solely on the basis of concept and relation names, not taking into account logical operators. Recently, the idea of reducing concept similarity to instance similarity has also been proposed by [7].

7 Summary and Conclusions

We presented a semantic similarity measure for DL-lite that builds on top of a similarity measure for A-Boxes that uses ideas [3]. We extend this measure to include T-Box information in terms of positive and negative inclusion statements. We illustrated the use of the similarity measure using a running example. The semantic nature of the measures makes it an ideal basis for improving semantic matching methods for heterogeneous ontologies and conceptual schemas. In future work, we will investigate the use of the measure in such scenarios. Further, we will investigate, if the principles underlying our similarity measure can also be applied to more expressive ontology languages.

References

1. Alex Borgida, Thomas J. Walsh, and Haym Hirsh. Towards measuring similarity in description logics. In *Proceedings of the 2005 International Workshop on Description Logics (DL2005)*, Edinburgh, Scotland, 2005.
2. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *Journal of Automated Reasoning*, 39:385–429, 2007.
3. Pierre-Antoine Champin and Christine Solnon. Measuring the similarity of labeled graphs. In *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 2003)*, pages 80–95, Trondheim, Norway, 2003.
4. C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. In *CILC 2005, Convegno Italiano di Logica Computazionale*, Rome, Italy, 2005.
5. C. D'Amato, S. Staab, and N. Fanizzi. On the influence of description logics ontologies on conceptual similarity. In *International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW 2008)*, Acitrezza, Catania, Italy, September/October 2008.
6. Bo Hu, Yannis Kalfoglou, David Dupplaw, Harith Alani, Paul Lewis, and Nigel Shadbolt. Semantic metrics. In *Proceedings 15 International Conference on Knowledge Engineering and Knowledge Management (EKAW06)*, Podebrady, Czech Rep., 2006.
7. Krysztof Janowicz and Marc Wilkes. SIM-DL$_a$: A novel semantic similarity measure reducing inter-concept to inter-instance similarity. In *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 353–367. Springer, 2009.
8. D. Maier, A.0. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Transactions on Database Systems*, 4:455–469, 1979.