SIM-DL$_{\text{A}}$: A Novel Semantic Similarity Measure for Description Logics Reducing Inter-concept to Inter-instance Similarity

Krzysztof Janowicz and Marc Wilkes

Institute for Geoinformatics, University of Muenster, Germany
{janowicz,marc.wilkes}@uni-muenster.de

Abstract. While semantic similarity plays a crucial role for human categorization and reasoning, computational similarity measures have also been applied to fields such as semantics-based information retrieval or ontology engineering. Several measures have been developed to compare concepts specified in various description logics. In most cases, these measures are either structural or require a populated ontology. Structural measures fail with an increasing expressivity of the used description logic, while several ontologies, e.g., geographic feature type ontologies, are not populated at all. In this paper, we present an approach to reduce inter-concept to inter-instance similarity and thereby avoid the canonization problem of structural measures. The novel approach, called SIM-DL$_{\text{A}}$, reuses existing similarity functions such as co-occurrence or network measures from our previous SIM-DL measure. The required instances for comparison are derived from the completion tree of a slightly modified DL-tableau algorithm as used for satisfiability checking. Instead of trying to find one (clash-free) model, the new algorithm generates a set of proxy individuals used for comparison. The paper presents the algorithm, alignment matrix, and similarity functions as well as a detailed example.

1 Introduction and Motivation

Semantic similarity measurement plays an important role in information retrieval on the semantic Web [1]. It supports users in searching and browsing through structured data. To improve human computer interaction, i.e., to be successfully integrated into semantics-aware user interfaces, such measures need to fulfill two criteria. First, they need to be able to handle the expressivity of the used description logics. Second, the similarity rankings produced by these measures have to correlate with human similarity rankings for the same set of compared concepts or individuals. In our previous work [2,3], we developed a structural similarity measure (SIM-DL[1]) for information retrieval on the semantic geospatial Web which fulfills these requirements [4]. It defines similarity as measure of conceptual overlap. As SIM-DL compares concept definitions, it

[1] The SIM-DL similarity server and Protégé plug-in are free and open-source software and can be downloaded at http://sim-dl.sourceforge.net/downloads/.

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 353–367, 2009.
© Springer-Verlag Berlin Heidelberg 2009

does not require a populated ontology. This is especially important for various applications in geoinformation science (GIScience) such as Web gazetteers. In addition to simple place name queries, gazetteers also support type queries (such as for *archaeological sites in Crete*). In the past, thesauri have been used for the definition and retrieval of these types. Advanced gazetteer interfaces support semantics-based retrieval and rely on geographic feature type ontologies [3]. For various reasons not discussed here in detail, the millions of geographic places defined over the last decades cannot be directly mapped to instances within such ontologies. While a semi-automatic mapping is part of the long term agenda towards a semantic geospatial Web, many geographic feature type ontologies remain unpopulated until now. As SIM-DL is a structural measure, its major shortcoming is the need for canonization, i.e., concept definitions have to be rewritten to a common form to eliminate syntactic influence. Hence, SIM-DL fails to handle expressive knowledge representation languages such as OWL. In this paper, we propose a novel measure called SIM-DL$_{\mathrm{A}}$. It overcomes these difficulties by reducing inter-concept to inter-instance similarity and reuses the similarity functions defined for SIM-DL. Our approach is based on a modified tableau algorithm which generates a set of proxy individuals for comparison.

2 Semantic Similarity Measurement

The theory of similarity has its roots in philosophy and psychology and was established to determine why and how entities are grouped into categories, and why some categories are comparable to each other while others are not [5,6]. Nowadays, similarity measures play an important role for ontology engineering, alignment and matching [7,8,9,10], as well as for information retrieval [3,11]. The main challenge for *semantic* similarity measurement is the comparison of meanings as opposed to a purely structural (syntactical) comparison. Depending on the representation language, concepts are specified as collections of features [12], regions in a multidimensional space [13], or formal restrictions specified on sets using description logics [14,15,16,2]. Besides representation, context is a major challenge for similarity. In most cases, meaningful measures cannot be defined without specifying a context in which similarity is measured [6,17,18,19,20].

2.1 Semantics of Similarity

As argued by Goodman [21], there is no global and application independent law on how similarity is measured. Strictly speaking, there is even no single definition of what similarity measures [6]. This makes the selection of an appropriate measure for a particular application area and also the comparison of existing similarity measures difficult. In the past, we have examined several measures for different applications and found generic patterns which jointly form a framework explaining how (inter-concept) similarity is measured [3]. It consists of the following seven steps. Their concrete realization depends on the similarity measure and the used representation language. While some of these steps are important for a particular measure, they may play a marginal role for another.

1. Definition of application area and intended audience
2. Selection of search (query) and target concepts
3. Transformation of concepts to canonical form
4. Definition of an alignment matrix for concept descriptors
5. Application of constructor specific similarity functions
6. Determination of standardized overall similarity
7. Interpretation of the resulting similarity value(s)

Every similarity measure should define in which way it implements the proposed steps and thereby specifies the semantics of similarity (values) as well as its properties, i.e., whether it is reflexive, symmetric, transitive, strict, minimal, etc.; see [22,23,5] for a detailed discussion. It is interesting to note that researchers from both cognitive science and artificial intelligence claim that most of these properties contradict with the nature of (human) similarity judgments. Finally, the framework allows for a better separation between the process of measuring similarity (i.e., what is measured) and the used similarity functions (i.e., how it is measured). In the following, a brief description of the steps is given.

Application Area and Intended Audience. While similarity measurement is not restricted to solve a particular task, most similarity functions have been developed for a specific purpose. Therefore, the question which functions should be selected depends on the application area. Theories developed for information retrieval and in cognitive science tend to use asymmetric similarity functions to explaining human similarity reasoning [5]. In contrast, measures developed for ontology alignment prefer symmetric functions (as there is no specific role between compared ontologies). Additionally, in case of disjunction one can choose between computing maximum, minimum [24], or average similarity [3]. Finally, human similarity judgments are influenced by age, language, and cultural background which may play a crucial role in human computer interaction [6,18].

Search and Target Concepts. Before similarity can be measured the compared concepts have to be selected. Depending on the application area, the search (or query) concept C_s can be either part of the examined ontology or phrased using a shared vocabulary [2,3]. The target concepts $C_{t_1}, ..., C_{t_i}$ form the so-called context of discourse $\mathcal{C}_d$ [18] and are selected by hand or automatically determined by specifying a context concept C_c. In the latter case, the target concepts are all concepts subsumed by C_c ($\mathcal{C}_d = \{C_t \mid C_t \sqsubseteq C_c\}$). The distinction between search and target concept is especially important for asymmetric similarity. The selection of a context concept does not only determine which concepts are compared, it also affects the measured similarity (see section 3.4). Based on search, target, and context concept similarity queries may look like the following ones:

- How similar is *Canal* (C_s) to *River* (C_t)?
- Which kind of *Waterbody* (C_c) is most similar to *Canal* (C_s)?

In the first case, *Canal* would be compared to *River*, while it is compared to all subconcepts of *Waterbody* (e.g., *River*, *Lake*, *Reservoir*) in the second case.

Canonical Form. Semantic similarity should depend on what is said about concepts, not how it is said. If two concept descriptions (specified in a given language) denote the same facts using different language elements, they need to be rewritten to a common form to eliminate unintended syntactic influences. This step mainly depends on the underlying representation language and is most important for structural similarity measures. A simple examples is:

$$\textbf{Condition } \forall R.C \sqcap \forall R.C' \ \textbf{ Rewrite } \forall R.C \sqcap \forall R.C' \text{ to } \forall R.(C \sqcap C')$$

The complexity of such rewriting rules increases with the expressivity of the used language and is a major obstacle in defining structural measures for concepts phrased in high expressive description logics.

Alignment Matrix. While the first steps of the framework determine which concepts are selected for comparison, the alignment matrix specifies which and how concept descriptors (e.g., dimensions, features, super/subconcepts) are compared. The term alignment is chosen here following research from psychology which investigates how structure and correspondence influences similarity judgments [25,6]. For instance, $\exists overlaps.C$ can be compared to $\exists inside.C$ because the involved roles are part of the same (topological) conceptual neighborhood [26]. In contrast, both cannot be compared to $\exists likes.C$. Likewise, the green wheel of a car cannot be aligned for comparison to the green hood of a truck to increase their similarity [5]. The term matrix points out that the selection of comparable tuples of descriptors requires a matrix $C_{Ds} \times C_{Dt}$ (where C_{Ds} and C_{Dt} are the sets of descriptors forming C_s, and C_t, respectively).

Similarity Functions. After selecting the compared concepts and aligning their descriptors, similarity is measured for each selected tuple. Depending on the used representation language different similarity functions have to be applied. For instance, in case of the Matching Distance Similarity Measure (MDSM) [12], features are distinguished into different types during the alignment process: parts, attributes, and functions. Although a context weighting is computed for each of these types, the same similarity function is applied to all of them. SIM-DL distinguishes between several similarity functions for roles and their fillers, e.g., functions for conceptual neighborhoods, role hierarchies, or co-occurrence of primitives. While we focus on inter-concept similarity here, some similarity functions also take individuals into account [15,24,20]. In most cases, each similarity function takes care of standardization (to values between 0 and 1) itself.

Overall Similarity. Next, the single similarity values derived from applying the similarity functions to all selected tuples (of the compared concepts) are combined to an overall similarity. In most theories this step is implemented as weighted summation function. For MDSM, the overall similarity is the weighted sum of the similarities determined for functions, parts, and attributes. The weights ω_f, ω_p, and ω_a indicate the relative importance of each feature type using either a commonality or variability model [12]. At the same time, the weights act as standardization factors ($\sum \omega = 1$). For conceptual space-based approaches, the overall similarity is given by the normalized, i.e., z-transformed sum of compared values [13]. In case of SIM-DL, the overall similarity is simply the sum of the similarities computed for the compared tuples.

Interpretation. Finally, a single similarity value is difficult to interpret. For instance, it does not answer the question whether there are more or less similar target concepts in the examined ontology, i.e., the distribution of similarities within the ontology is unknown. Moreover, an isolated comparison puts too much stress on the specific similarity value. It is difficult to argue that and why the result is (cognitively) plausible without other reference values. Therefore, SIM-DL focuses on similarity rankings visualized as lists of descending similarities, font-size scaling as known from tag clouds, or the clustering of the results.

3 The Novel SIM-DL$_{\text{A}}$-Measure

In this section, the novel SIM-DL$_{\text{A}}$-Measure is introduced. It reduces inter-concept to inter-instance similarity to avoid the problem of canonization known from purely structural measures. As SIM-DL$_{\text{A}}$ should not require a populated ontology, the instances have to be generated before similarity is measured. This is achieved by creating a representative set of instances for both the search and target concept(s) using a slightly modified tableau algorithm as known from satisfiability checking. The algorithm produces additional models, if necessary, in order to account for all possible interpretations of a concept. To be tailored to the user's needs and application area (see step one of the framework), SIM-DL$_{\text{A}}$ supports symmetric and asymmetric versions of maximum and minimum similarity. For reasons of simplification, i.e., to avoid substraction operations [27] on DL concepts, the context concept is restricted to primitive concepts. In this paper, we discuss the modified tableau algorithm for the description logic $\mathcal{SHI}$ and focus on step three to six of the similarity framework. Hence, we start with the alignment matrix and introduce the applied similarity functions afterwards. A detailed example is given in section 4.

3.1 Syntax and Semantics of $\mathcal{SHI}$

Primitive concepts and roles can be combined using constructors to build complex concepts. The language $\mathcal{SHI}$ provides the constructors *intersection*, *union existential quantification*, *value restriction*, and *complex concept negation* (table 1). In addition, roles can be defined to be *transitive*, *inverse* or form a *role hierarchy*. In the following, the letters C, D represent concepts and R, S roles. N_C denotes the set of concept names, accordingly N_R represents the set of role names. N_R^+ is the subset of N_R which contains only transitive roles. Finally, the letters x, y, and z represent individuals. The formal semantics of a $\mathcal{SHI}$-concept is given by its interpretation $\mathcal{I}$. $\mathcal{I}$ consists of a non-empty set $\triangle^{\mathcal{I}}$, which is called the domain of interpretation, and an interpretation function which maps a concept name C to a set of individuals ($C^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$) and a role R to a set of pairs of individuals ($R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$). An interpretation that satisfies the axioms of a TBox $\mathcal{T}$ is called a *model* of $\mathcal{T}$. Accordingly, in the presence of an ABox $\mathcal{A}$, an interpretation is called a model of $\mathcal{A}$ if it satisfies $(x^{\mathcal{I}}, y^{\mathcal{I}}) \in R^{\mathcal{I}}$ for all role assertions $R(x, y) \in \mathcal{A}$, and $x^{\mathcal{I}} \in C^{\mathcal{I}}$ for all concept assertions $C(x) \in \mathcal{A}$.

Table 1. Syntax and semantics of $\mathcal{SHI}$

Name	Syntax	Semantics
Atomic concept	A	$A^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$
Atomic role	R	$R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$
Concept negation	$\neg C$	$\triangle^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Concept intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Concept union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Existential quantification	$\exists R.C$	$\{x \in \triangle^{\mathcal{I}} \mid \exists y.(x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Value restriction	$\forall R.C$	$\{x \in \triangle^{\mathcal{I}} \mid \forall y.(x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
Inverse role	R^{-}	$\{(x, y) \in \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}} \mid (y, x) \in R^{\mathcal{I}}\}$
Transitive roles	$R \in N_R^{+}$	$\{(x, z) \mid (x, y) \in R^{\mathcal{I}} \wedge (y, z) \in R^{\mathcal{I}}\}$
Role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

3.2 Similarity Tableau for $\mathcal{SHI}$

The tableau algorithm introduced in this section is part of the fourth step of the similarity framework – the definition of an alignment matrix. Before the matrix can be constructed, the third step requires the concepts to be rewritten to a canonical form. Sim-DL$_{\text{A}}$ is designed to avoid a complex canonization, i.e., the concepts are simply rewritten to *negation normal form (NNF)*. Additionally, concept inclusion is rewritten to equivalence using the *primitiveness* of concepts [28]. For example, $C \sqsubseteq A$ can be rewritten as $C \equiv A \sqcup C'$.

Tableau algorithms prove the satisfiability of a concept C by trying to construct a model of C. More precisely, as soon as the algorithm constructed *one* model of C the satisfiability is affirmed and the algorithm terminates. In contrast, the expansion rules presented in this section do not aim at constructing only one model of C, but rather a number of models which act as proxies for the extension of C. These models are called *proxy models* here. In the following, the tableau expansion rules are presented. Since these rules differ only slightly from those used for checking satisfiability, readers interested in more details are referred to the work of Horrocks et al. [29,30].

The application of the expansion rules yields a *completion tree* $\mathbf{T}$. For a concept C, this is a tree where each node is labeled with a set $\mathcal{L}(x)$ of concept expressions occurring in C. An edge $\langle x, y\rangle$ is either labeled with $\sqcup$ or $\forall$ (and indicates another potential proxy model starting at the y-node), or with $\mathcal{L}(\langle x, y\rangle) = R$, where R is a role occurring in C [29]. For a node x, $\mathcal{L}(x)$ contains a *clash* if, for some concept C, $\{C, \neg C\} \subseteq \mathcal{L}(x)$. Those branches of the tree $\mathbf{T}$ which do not contain a clash are proxy models and enter the similarity measurement process. A node y is called a R-neighbor of x if either y is a *successor* of x (i.e., x and y are connected by an edge $\langle x, y\rangle$) and $\mathcal{L}(\langle x, y\rangle) = S$ or y is a *predecessor* of x (x and y are connected by an edge $\langle y, x \rangle$) and $\mathcal{L}(\langle y, y \rangle) = Inv(S)$[2] for some S within the transitive closure of R ($S \sqsubseteq^* R$). A node x is *blocked* if one of the *ancestors* is blocked or $\mathcal{L}(x) = \mathcal{L}(y)$, where *ancestor* is the transitive closure of *predecessor*.

In contrast to the original tableau algorithm [30], two expansion rules are modified. First, the $\sqcup$-rule is modified in order to generate a third possible new model allowing an individual for being an instance of both concepts participating in the union. Second, the $\forall$-rule is modified. In the absence of $\exists$-expressions, the modified $\forall$-rule generates two possible models. One of these models will be unchanged (as in standard tableau algorithms), the other will be enriched by an existential quantification. When a value restriction was explicitly specified for a given concept, this should impact similarity and hence has to be reflected by a proxy model. The following expansions rules are applied:

The $\sqcap$-rule:
Condition: $C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and $\{C_1, C_2\} \notin \mathcal{L}(x)$.
Action: $\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_1, C_2\}$.

The $\sqcup$-rule:
Condition: $C_1 \sqcup C_2 \in \mathcal{L}(x)$ and x is not indirectly blocked.
Action: Create three $\sqcup$-successors w, y, z of x with:
$\mathcal{L}(w) := (\mathcal{L}(x) \backslash \{C_1 \sqcup C_2\}) \cup \{C_1\}$
$\mathcal{L}(y) := (\mathcal{L}(x) \backslash \{C_1 \sqcup C_2\}) \cup \{C_2\}$
$\mathcal{L}(z) := (\mathcal{L}(x) \backslash \{C_1 \sqcup C_2\}) \cup \{C_1, C_2\}$

The $\exists$-rule:
Condition: $(\exists R.C) \in \mathcal{L}(x)$, x is not blocked, and x has no R-neighbor y with $C \in \mathcal{L}(y)$.
Action: Create a new node y with $\mathcal{L}(y) = \{C\}$ and the edge $\mathcal{L}(\langle x, y \rangle) = R$.

The $\forall$-rule:
Condition: $(\forall R.C) \in \mathcal{L}(x)$, x is not indirectly blocked.
Action:
If there is an R-neighbor y of x and $C \notin \mathcal{L}(y)$:
$\mathcal{L}(y) := \mathcal{L}(y) \cup \{C\}$.
If there is no R-neighbor y of x, create two $\forall$-successors y, z of x with:
$\mathcal{L}(y) := \mathcal{L}(x)$ (y will then be blocked)
$\mathcal{L}(z) := \mathcal{L}(x) \cup \{\exists R.C\}$.

The $\forall_+$-rule:
Condition: $(\forall R.C) \in \mathcal{L}(x)$, x is not indirectly blocked, there is some S which is transitive and $S \sqsubseteq R$, and there is an S-neighbor y of x with $\forall S.C \notin \mathcal{L}(y)$.
Action: $\mathcal{L}(y) := \mathcal{L}(y) \cup \{\forall S.C\}$.

3.3 Alignment Matrices

For a search concept C_s and a target concept C_t, the completion tree obtained from applying the modified tableau algorithm serves as starting point to derive

[2] $Inv(S) := S^-$, if S is a role name. $Inv(S) := R$, if $S = R^-$ for a role name R [30].

the alignment matrices (the fourth step of the framework). We distinguish two levels of matrices, the model level and assertion level matrices.

Model Level Matrix – Selecting Models for Comparison. For C_s and C_t, each completion tree contains a set of proxy models. These proxy models define the matrix M_M (where $_M$ indicates that models are compared). More precisely, the number of columns is determined by the number of models created for the search concept. Accordingly, each model created for the target concept corresponds to one row in the matrix. For each field in M_M, the similarity between the two models is computed and entered into the matrix. Which of these tuples are selected for comparison depends on the application and user. For example, if an application requires maximum similarity, the tuple with the highest value is returned. For minimum similarity, the tuple with the lowest similarity is selected.

Assertion Level Matrix – Computing Similarity between Models. While the first matrix is concerned with selecting appropriate models for comparison, the following matrix M_A (where $_A$ indicates that assertions are compared) addresses the comparison of two actual instances. Therefore, each primitive concept that is instantiated by a model's root individual is added as a column/row to the matrix. Furthermore, each outgoing edge (representing a role assertion) is added as well. Now, the similarity between two primitive concepts can be computed. The similarity between role assertions consists of two steps. First, the similarity between the roles is computed. The role similarity determines which role assertions are aligned, i.e., filler similarity is only computed when appropriate. If the filler similarity needs to be determined, another assertion level alignment matrix for the two filler individuals is created. Once the matrix is filled, those pairs of assertions with the highest similarity values are selected for comparison. It is important to note that each column and each row is only selected once, avoiding a too heavy influence of single assertions. The similarity values are summed up and standardized. The standardization factor σ (see equation (4)) depends on whether the symmetric or asymmetric measure is chosen [2,3]. In the latter case, σ is the number of assertions of the search individual (i.e., the number of columns). In the symmetric case, the maximum number of either columns or rows is used.

With respect to step one of the framework, the distinction between maximum and minimum similarity is reflected in the model level matrix. It determines which of the proxy models are compared. Symmetry and asymmetry are reflected in the assertion level matrix which selects assertions for comparison. This way, and in contrast to measures such as MDSM [12], SIM-DL$_{\mathrm{A}}$ can be tailored to a specific application area without altering the similarity functions themselves.

3.4 Measuring Inter-instance Similarity

In this section, we introduce the concrete similarity functions used to determine the tuples selected for comparison. Each assertion is either of the form $A(x)$ or $r(x,y)$, where A is a primitive concept instantiated by x and $r(x,y)$ is an ordered pair of individuals in $R^{\mathcal{I}}$. Following the introduced framework, three similarity functions are necessary: one for primitives, roles, and role-filler-pairs.

Primitive Concepts. Primitives (base symbols) occur only on the right-hand side of axioms, i.e., they are not definable. To measure similarity between primitives (sim_p; see equation (1)), an adapted version of the Jaccard Similarity Coefficient is used. It measures the degree of overlap between two sets S_1 and S_2 as ratio of the cardinality of shared members from $S_1 \wedge S_2$ to the cardinality retrieved from $S_1 \vee S_2$. MDSM [12] uses an asymmetric version of Jaccard's coefficient, while in case of SIM-DL [3] it is adapted to compute the context-aware co-occurrence of primitives within the definitions of other (non-primitive) concepts. Two primitives are the more similar, the more complex concepts are defined using both (and not only one) of them. If $sim_p(A,B) = 1$, both primitives always co-occur in complex concepts and cannot be distinguished. Hence, to measure the similarity between assertions of the type $A(x)$ and $B(y)$, the similarity between A and B is measured. As similarity also depends on the context of discourse [18], we only consider those concepts C_i which are subconcepts of C_c (see step two of the similarity framework).

$$sim(A(x), B(y)) = sim_p(A,B) = \frac{\mid \{C \mid (C \sqsubseteq C_c) \wedge (C \sqsubset A) \wedge (C \sqsubset B)\} \mid}{\mid \{C \mid (C \sqsubseteq C_c) \wedge ((C \sqsubset A) \vee (C \sqsubset B))\} \mid} \tag{1}$$

Role Hierarchy. $\mathcal{SHI}$ supports role hierarchies, i.e., role inclusion, but does not support intersection or composition. Same as argued for primitives, there are no role definitions which can be compared for similarity. Because of the missing intersection constructor we cannot apply Jaccard's coefficient here. Instead, a network-based approach [31] is taken to compute the similarity of roles (R and S) within a hierarchy. Similarity (sim_r; see equation (2)) is defined as ratio between the shortest path from R to S and the maximum path within the graph representation of the role hierarchy; where the universal role U ($U \equiv \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$) forms the graph's root. Compared to sim_p, similarity between roles is defined without reference to the context. This would require to take only such roles into account which are used within quantifications or restrictions of concepts within the context. The standardization in equation (2) is depth-dependent to indicate that the distance from node to node decreases with increasing depth level of R and S within the hierarchy. In other words, the weights of the edges used to determine the path between R and S decrease with increasing depth of the graph. If a path between two roles crosses U, similarity is 0. The $lcs(R,S)$ is the least common subsumer, in this case the first common super role of R and S.

$$sim_r(R,S) = \frac{depth(lcs(R,S))}{depth(lcs(R,S)) + edge\_distance(R,S)} \tag{2}$$

SIM-DL uses a second function to compare roles within a conceptual neighborhood [2]. This is necessary for topological and temporal relations as used in GIScience[3]. One could also define network-based similarity functions for other roles such as *part-of*. These functions can be integrated into SIM-DL$_{\mathrm{A}}$.

[3] See [32] on the problems of integrating topological relations and reasoning in DL.

Roles and Fillers. The similarity between assertions of the type $r(x, y)$ and $s(w, z)$, is the similarity of the involved roles R and S times the overall similarity of the assertions about y and z; where x and w are the individuals to be compared (see equation (3)).

$$sim(r(x, y), s(w, z)) = sim_r(R, S) * sim_o(y, z) \tag{3}$$

Some similarity measures define role-filler similarity as weighted average of the role and filler similarities, but the multiplicative approach has turned out to be cognitively plausible [4] and allows for simple approximation and optimization techniques not discussed here in detail.

3.5 Overall Similarity

As sim_p and sim_r deliver similarity values between 0 and 1, the overall instance similarity (sim_o; see equation (4)) in SIM-DL$_\mathrm{A}$ is simply the standardized sum of the similarities computed for all assertion tuples selected during the alignment; see section 3.3. The similarity of compared concepts is either the similarity of the most or least similar proxy individuals (computed using sim_o).

$$sim_o(x, y) = \frac{\sum M_{A_{ij}}}{\sigma}; \text{ where } M_{A_{ij}} \text{ is a selected tuple from the matrix } M_A. \tag{4}$$

4 Exemplary Application of SIM-DL$_\mathrm{A}$

The previous section presented the novel SIM-DL$_\mathrm{A}$ measure. In order to illustrate the steps involved in a comparison process between a search concept C_s and a target concept C_t, this section provides a detailed example. A, B, D, and E are primitive concepts, R and S are roles with $R \sqsubset S$.

$C_s \equiv A \sqcap (B \sqcup D) \sqcap \exists R.E$ $\qquad$ $C_t \equiv A \sqcap (B \sqcup \neg A) \sqcap \forall S.E \sqcap \forall S.D$

For C_s, the expansion rules are applied as follows, the expansion tree $\mathbf{T}_s$ and resulting models are shown in figure 1.

1. Unfold C_s and convert it to NNF.	$A \sqcap (B \sqcup D) \sqcap \exists R.E$
2. Initialize $\mathbf{T}_s$ containing a node s_0.	$\mathcal{L}(s_0) = \{A \sqcap (B \sqcup D) \sqcap \exists R.E\}$
3. Apply the $\sqcap$-rule to $A \sqcap (B \sqcup D) \sqcap \exists R.E \in \mathcal{L}(s_0)$.	$\mathcal{L}(s_0) := \mathcal{L}(s_0) \cup \{A, B \sqcup D, \exists R.E\}$
4. Apply the $\sqcup$-rule to $B \sqcup D \in \mathcal{L}(s_0)$. Create three $\sqcup$-successors s_1, s_2, s_3 of s_0.	$\mathcal{L}(s_1) := \mathcal{L}(s_0) \backslash \{B \sqcup D\} \cup \{B\}$ $\mathcal{L}(s_2) := \mathcal{L}(s_0) \backslash \{B \sqcup D\} \cup \{D\}$ $\mathcal{L}(s_3) := \mathcal{L}(s_0) \backslash \{B \sqcup D\} \cup \{B, D\}$
5. Apply the $\exists$-rule to $\exists R.E \in \mathcal{L}(s_1)$. Create a node s_4 and an edge $\langle s_1, s_4 \rangle$.	$\mathcal{L}(s_4) = \{E\}$ $\mathcal{L}(\langle s_1, s_4 \rangle) = \{R\}$
6. Apply the $\exists$-rule to $\exists R.E \in \mathcal{L}(s_2)$. Create a node s_5 and an edge $\langle s_2, s_5 \rangle$.	$\mathcal{L}(s_5) = \{E\}$ $\mathcal{L}(\langle s_2, s_5 \rangle) = \{R\}$
7. Apply the $\exists$-rule to $\exists R.E \in \mathcal{L}(s_3)$. Create a node s_6 and an edge $\langle s_3, s_6 \rangle$.	$\mathcal{L}(s_6) = \{E\}$ $\mathcal{L}(\langle s_3, s_6 \rangle) = \{R\}$

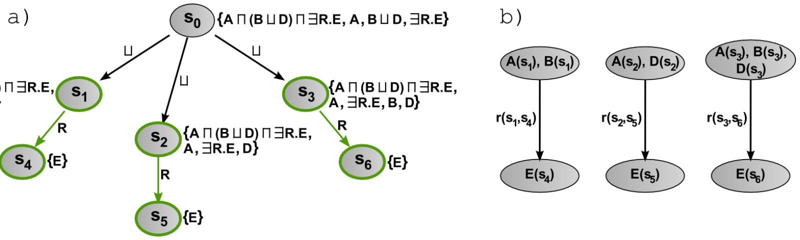

a)
s0 {A ⊓ (B ⊔ D) ⊓ ∃R.E, A, B ⊔ D, ∃R.E}
⊔
⊔
⊔
{A ⊓ (B ⊔ D) ⊓ ∃R.E, A, ∃R.E, B} s1
s2 {A ⊓ (B ⊔ D) ⊓ ∃R.E, A, ∃R.E, D}
s3 {A ⊓ (B ⊔ D) ⊓ ∃R.E, A, ∃R.E, B, D}
R
R
R
s4 {E}
s5 {E}
s6 {E}
b)
A(s1), B(s1)
r(s1,s4)
E(s4)
A(s2), D(s2)
r(s2,s5)
E(s5)
A(s3), B(s3), D(s3)
r(s3,s6)
E(s6)

Fig. 1. a) Expansion tree for C_s. b) Models for C_s.

For C_t, the expansion tree $\mathbf{T}_t$ is slightly more complex. It results from applying the expansion rules as listed below. The tree as well as the models are shown in figure 2.

1. Unfold C_t and convert it to NNF.	$A \sqcap (B \sqcup \neg A) \sqcap \forall S.E \sqcap \forall S.D$
2. Initialize $\mathbf{T}_t$ containing a node t_0.	$\mathcal{L}(t_0) = \{A \sqcap (B \sqcup \neg A) \sqcap \forall S.E \sqcap \forall S.D\}$
3. Apply the $\sqcap$-rule to $A \sqcap (B \sqcup \neg A) \sqcap \forall S.E \sqcap \forall S.D \in \mathcal{L}(t_0)$.	$\mathcal{L}(t_0) := \mathcal{L}(t_0) \cup \{A, B \sqcup \neg A, \forall S.E, \forall S.D\}$
4. Apply the $\sqcup$-rule to $B \sqcup \neg A \in \mathcal{L}(t_0)$. Create three $\sqcup$-successors t_1, t_2, t_3 of t_0.	$\mathcal{L}(t_1) := \mathcal{L}(t_0) \setminus \{B \sqcup \neg A\} \cup \{B\}$ $\mathcal{L}(t_2) := \mathcal{L}(t_0) \setminus \{B \sqcup \neg A\} \cup \{\neg A\}$ (clash) $\mathcal{L}(t_3) := \mathcal{L}(t_0) \setminus \{B \sqcup \neg A\} \cup \{B, \neg A\}$ (clash)
5. Apply the $\forall$-rule to $\forall S.E \in \mathcal{L}(t_1)$. Create two $\forall$-successors t_4, t_5 of t_1.	$\mathcal{L}(t_4) := \mathcal{L}(t_1)$ (t_4 is blocked by t_1) $\mathcal{L}(t_5) := \mathcal{L}(t_1) \cup \{\exists S.E\}$
6. Apply the $\exists$-rule to $\exists S.E \in \mathcal{L}(t_5)$. Create a node t_6 and an edge $\langle t_5, t_6 \rangle$.	$\mathcal{L}(t_6) = \{E\}$ $\mathcal{L}(\langle t_5, t_6 \rangle) = \{S\}$
7. Apply the $\forall$-rule to $\forall S.D \in \mathcal{L}(t_5)$. There is a S-neighbor t_6 of t_5, but $D \notin \mathcal{L}(t_6)$.	$\mathcal{L}(t_6) := \mathcal{L}(t_6) \cup \{D\}$

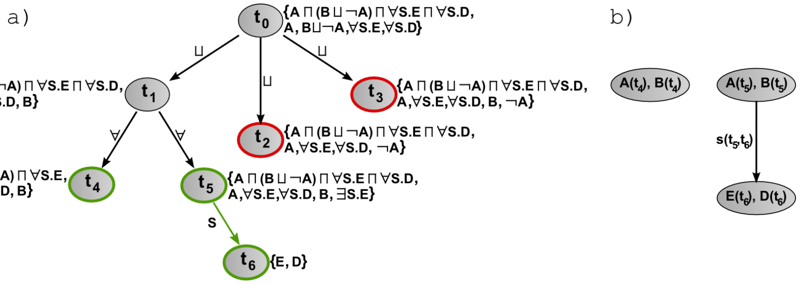
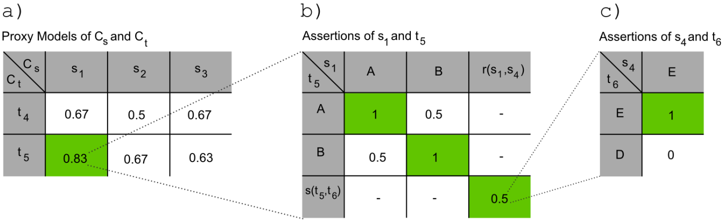

a)
t0 {A ⊓ (B ⊔ ¬A) ⊓ ∀S.E ⊓ ∀S.D, A, B⊔¬A,∀S.E,∀S.D}
⊔
⊔
⊔
{A ⊓ (B ⊔ ¬A) ⊓ ∀S.E ⊓ ∀S.D, A,∀S.E,∀S.D, B} t1
t3 {A ⊓ (B ⊔ ¬A) ⊓ ∀S.E ⊓ ∀S.D, A,∀S.E,∀S.D, B, ¬A}
t2 {A ⊓ (B ⊔ ¬A) ⊓ ∀S.E ⊓ ∀S.D, A,∀S.E,∀S.D, ¬A}
∀
∀
{A ⊓ (B ⊔ ¬A) ⊓ ∀S.E, A,∀S.E,∀S.D, B} t4
t5 {A ⊓ (B ⊔ ¬A) ⊓ ∀S.E ⊓ ∀S.D, A,∀S.E,∀S.D, B, ∃S.E}
S
t6 {E, D}
b)
A(t4), B(t4)
A(t5), B(t5)
s(t5,t6)
E(t6), D(t6)

Fig. 2. a) Expansion tree for C_t. b) Models for C_t.

The application of the tableau algorithm generates three proxy models for C_s and two for C_t. As shown in figure 3a), the similarity between each possible tuple of models is computed. In the model level matrix, a model is represented by its root node. Given that the user queries for the maximum similarity between C_s and C_t, $M_{M_{12}}$ is selected and yields the similarity value 0.83. The assertion level matrix is shown in figure 3b). As described in section 3.3, each concept and role assertion is added as a column (or row, respectively) in the alignment matrix. For simplification we assume that $C_c = \top$ and $sim(A, B) = 0.5$. Whereas the similarity between primitive concepts can directly be computed using function (1), those fields in the alignment matrix representing tuples of role assertions are computed using similarity function (3), hence demanding for two calculations, $sim_r(R, S)$ and $sim_o(s_4, t_6)$. The former function calculates the similarity between two roles. R is a direct subrole of S and for simplification we assume that $sim(R, S)$ yields 0.5. As the role similarity is above 0 and there are no other role assertion tuples, the filler similarity is computed. To do so, another alignment matrix for s_4 and t_6 is created, resulting in $sim_o(s_4, t_6) = 1$ (see figure 3c).



a)

Proxy Models of C_s and C_t

C_s / C_t	s_1	s_2	s_3
t_4	0.67	0.5	0.67
t_5	0.83	0.67	0.63

b)

Assertions of s_1 and t_5

s_1 / t_5	A	B	$r(s_1,s_4)$
A	1	0.5	-
B	0.5	1	-
$s(t_5,t_6)$	-	-	0.5

c)

Assertions of s_4 and t_6

s_4 / t_6	E
E	1
D	0

Fig. 3. The alignment matrices for comparing C_s and C_t. Given that asymmetric maximum similarity is chosen, the green fields show the tuples selected for comparison.

5 Conclusions and Further Work

In this paper, we introduced the novel Sim-DL$_{\mathrm{A}}$ measure which reduces inter-concept to inter-instance similarity by generating proxy models using a modified tableau algorithm for $\mathcal{SHI}$. The algorithm can still be used for satisfiability checking, although its performance decreases with an extensive use of disjunctions and value restrictions, since additional models have to be created. Consequently, it should be implemented as optional *completion strategy* as known from the Pellet reasoner. The complexity of Sim-DL$_{\mathrm{A}}$ depends on the used DL (i.e., the tableau rules) and the complexity of computing the similarity for the matrices. Nevertheless, structural measures would require to define and implement a complete set of canonization rules. Following the introduced similarity framework, Sim-DL$_{\mathrm{A}}$ clearly separates similarity modes such as maximum/minimum similarity and symmetry/asymmetry, which are used to tailor the measure to the user's needs, from the concrete similarity functions. This allows to replace particular functions without changing the measurement process.

Our next steps involve the extension of the presented approach to more expressive description logics such as $\mathcal{SHIQ}$, i.e., to account for qualified number restrictions. We also work on the integration of Sim-DL$_A$ into the SIM-DL server and Protégé plug-in. In this paper we focused on the theory. While the new measure relies on the same similarity functions, the alignment process differs. Hence, we have to redo our human participants tests [4] to verify that the Sim-DL$_A$ similarity rankings correlate with human rankings. We also need to investigate how the creation of proxy individuals impacts similarity. Like the work of Araújo and Pinto [16] our measure could also be extended to compare ontologies. Finally, one could try to express similarity as the ratio of clashing versus clash-free models of compared concepts. The clashes can then be presented (in natural language) to the end-user to explain the resulting similarity values.

Acknowledgments

The comments from four anonymous reviewers as well as from Simon Scheider provided useful suggestions to improve the content and clarity of the paper. This work is funded by the *SimCat* project granted by the German Research Foundation (DFG Ra1062/2-1), as well as the International Research Training Group on *Semantic Integration of Geospatial Information* (DFG GRK 1498).

References

1. Rissland, E.L.: AI and similarity. IEEE Intelligent Systems 21(3), 39–49 (2006)
2. Janowicz, K.: Sim-DL: Towards a semantic similarity measurement theory for the description logic $\mathcal{ALCNR}$ in geographic information retrieval. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1681–1692. Springer, Heidelberg (2006)
3. Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., Baeumer, B.: Algorithm, implementation and application of the SIM-DL similarity server. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 128–145. Springer, Heidelberg (2007)
4. Janowicz, K., Keßler, C., Panov, I., Wilkes, M., Espeter, M., Schwarz, M.: A study on the cognitive plausibility of SIM-DL similarity rankings for geographic feature types. In: Bernard, L., Friis-Christensen, A., Pundt, H. (eds.) 11th AGILE International Conference on Geographic Information Science, Girona, Spain. Lecture Notes in Geoinformation and Cartography, pp. 115–133. Springer, Heidelberg (2008)
5. Goldstone, R.L., Son, J.: Similarity. In: Holyoak, K., Morrison, R. (eds.) Cambridge Handbook of Thinking and Reasoning, pp. 13–36. Cambridge University Press, Cambridge (2005)
6. Medin, D., Goldstone, R., Gentner, D.: Respects for similarity. Psychological Review 100(2), 254–278 (1993)
7. Cruz, I., Sunna, W.: Structural alignment methods with applications to geospatial ontologies. Transactions in GIS 12(6), 683–711 (2008)
8. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in OWL-lite. In: de Mántaras, R.L., Saitta, L. (eds.) Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), pp. 333–337. IOS Press, Amsterdam (2004)

9. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems: OTM 2008. LNCS, vol. 5332, pp. 1164–1182. Springer, Heidelberg (2008)
10. Falconer, S., Noy, N., Storey, M.A.: Ontology mapping - a user survey. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the Workshop on Ontology Matching (OM 2007) at ISWC/ASWC 2007, Busan, South Korea (2007)
11. Ricklefs, M., Blomqvist, E.: Ontology-based relevance assessment: An evaluation of different semantic similarity measures. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems: OTM 2008. OTM Conferences (2). LNCS, vol. 5332, pp. 1235–1252. Springer, Heidelberg (2008)
12. Rodríguez, A., Egenhofer, M.: Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. International Journal of Geographical Information Science 18(3), 229–256 (2004)
13. Raubal, M.: Formalizing conceptual spaces. In: Varzi, A., Vieu, L. (eds.) Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004), vol. 114, pp. 153–164. IOS Press, Torino (2004)
14. Borgida, A., Walsh, T., Hirsh, H.: Towards measuring similarity in description logics. In: International Workshop on Description Logics (DL 2005), July 2005. CEUR Workshop Proceedings, vol. 147. CEUR, Edinburgh (2005)
15. d'Amato, C., Fanizzi, N., Esposito, F.: A semantic similarity measure for expressive description logics. In: CILC 2005, Convegno Italiano di Logica Computazionale, Rome, Italy (2005)
16. Araújo, R., Pinto, H.S.: Semilarity: Towards a model-driven approach to similarity. In: International Workshop on Description Logics (DL 2007), vol. 20, pp. 155–162. Bolzano University Press, Bozen-Bolzano (2007)
17. Albertoni, R., Martino, M.D.: Semantic similarity of ontology instances tailored on the application context. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 1020–1038. Springer, Heidelberg (2006)
18. Janowicz, K.: Kinds of contexts and their impact on semantic similarity measurement. In: Proceedings of the 6th IEEE International Conference on Pervasive Computing and Communications; 5th Workshop on Context Modeling and Reasoning (CoMoRea 2008), Hong Kong, pp. 441–446. IEEE Computer Society, Los Alamitos (2008)
19. Keßler, C.: Similarity measurement in context. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS, vol. 4635, pp. 277–290. Springer, Heidelberg (2007)
20. d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: An inductive approach. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 288–302. Springer, Heidelberg (2008)
21. Goodman, N.: Seven strictures on similarity. In: Problems and projects, pp. 437–447. Bobbs-Merrill, New York (1972)
22. Ashby, F.G., Perrin, N.A.: Toward a unified theory of similarity and recognition. Psychological Review 95, 124–150 (1988)
23. Cross, V., Sudkamp, T.: Similarity and Computability in Fuzzy Set Theory: Assessments and Applications. Studies in Fuzziness and Soft Computing, vol. 93. Physica-Verlag (2002)
24. d'Amato, C., Fanizzi, N., Esposito, F.: A dissimilarity measure for $\mathcal{ALC}$ concept descriptions. In: SAC 2006: Proceedings of the 2006 ACM symposium on Applied computing, pp. 1695–1699. ACM Press, New York (2006)

25. Markman, A.B.: Structural alignment, similarity, and the internal structure of category representations. In: Similarity and Categorization, pp. 109–130. Oxford University Press, Oxford (2001)
26. Egenhofer, M., Al-Taha, K.: Reasoning about gradual changes of topological relationships. In: Frank, A.U., Formentini, U., Campari, I. (eds.) GIS 1992. LNCS, vol. 639, pp. 196–219. Springer, Heidelberg (1992)
27. Teege, G.: Making the difference: A subtraction operation for description logics. In: Doyle, J., Sandewall, E., Torasso, P. (eds.) 4th International Conference on Principles of Knowledge Representation and Reasoning (KR 1994), Bonn, Germany, pp. 540–550. Morgan Kaufmann, San Francisco (1994)
28. Horrocks, I.: Implementation and Optimization Techniques. In: The Description Logic Handbook: Theory, Implementation and Applications, pp. 306–346. Cambridge University Press, Cambridge (2003)
29. Horrocks, I.: Optimising Tableaux Decision Procedures for Description Logics. PhD thesis, University of Manchester (1997)
30. Horrocks, I., Sattler, U., Tobies, S.: A description logic with transitive and converse roles, role hierarchies and qualifying number restrictions. LTCS-Report LTCS-99-08, LuFG Theoretical Computer Science, RWTH Aachen (1999)
31. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics 19, 17–30 (1989)
32. Lutz, C., Möller, R.: Defined topological relations in description logics. In: Rousset, M.C., Brachman, R., Donini, F., Franconi, E., Horrocks, I., Levy, A. (eds.) Proceedings of the International Workshop on Description Logics, Gif sur Yvette, Paris, France, Université Paris-Sud, Centre d'Orsay, September 1997, pp. 15–19 (1997)