Leveraging Terminological Structure for Object Reconciliation

Jan Noessner, Mathias Niepert,
Christian Meilicke, and Heiner Stuckenschmidt

KR & KM Research Group
University of Mannheim, B6 26, 68159 Mannheim, Germany
{jan,mathias,christian,heiner}@informatik.uni-mannheim.de

Abstract. It has been argued that linked open data is the major benefit of semantic technologies for the web as it provides a huge amount of structured data that can be accessed in a more effective way than web pages. While linked open data avoids many problems connected with the use of expressive ontologies such as the knowledge acquisition bottleneck, data heterogeneity remains a challenging problem. In particular, identical objects may be referred to by different URIs in different data sets. Identifying such representations of the same object is called object reconciliation. In this paper, we propose a novel approach to object reconciliation that is based on an existing semantic similarity measure for linked data. We adapt the measure to the object reconciliation problem, present exact and approximate algorithms that efficiently implement the methods, and provide a systematic experimental evaluation based on a benchmark dataset. As our main result, we show that the use of light-weight ontologies and schema information significantly improves object reconciliation in the context of linked open data.

1 Introduction

There is an ongoing debate concerning the role of ontologies for the semantic web. While rich ontologies have been promoted as an integral part of every semantic web application [11], it is increasingly argued that the real value of the semantic web is based on its ability to create and maintain linked open data which provides effective access to semantically enhanced information on the web [21]. In this paper, we argue that the use of (light-weight) ontologies helps to solve one of the key problems of linked open data on the web, namely, the actual linking of data by identifying different representations of the same object. This problem has been extensively studied in the context of database systems as duplicate detection, record linkage, and object or reference reconciliation [13]. Most existing work has focused on the design of specialized measures which estimate the similarity of objects based on their lexical properties. The Silk framework [22], for instance, combines lexical similarity measures in order to create links between objects. The use of schema information in the context of formal ontologies has only recently been proposed [15, 10]. In this work, we leverage schema information to exclude logically inconsistent links between objects

and to improve the overall accuracy of instance alignments. In particular, we use logical reasoning and linear optimization techniques to compute the overlap of derivable types of objects. This information is combined with the classical similarity-based approach, resulting in a novel framework for object reconciliation. Our contributions are the following:

- We combine classical similarity measures for object reconciliation with a semantic similarity measure that takes schema information into account;
- We show that the combined approach clearly outperforms methods that do not consider schema information;
- We present efficient ways of computing the combined similarity measures based on a formulation as an integer linear programming problem; and
- We show that the method can be efficiently implemented using an approximate algorithm with only a modest loss of precision and recall.

The paper is organized as follows. In Section 2 we discuss the object reconciliation problem in more detail and refer to existing work in this area. Section 3 extends and adapts the similarity measure proposed in [19] to the problem of object reconciliation. In Section 4, we show that computing the maximal similarity between objects in two datasets can be formulated as an optimization problem. In particular, we show that there exists a transformation to a linear integer programming problem, the solution of which corresponds to the alignment that maximizes the semantic similarity between the datasets. In addition, we apply an existing approximative graph matching algorithm to the problem. Finally, in Section 5, we show that both the optimal and the approximate algorithms result in high-quality alignments both in terms of precision and recall.

2 Problem Statement and Related Work

The problem of object reconciliation has been a topic of research for more than 50 years. It is also known as the problem of record linkage [8], entity resolution [1], and instance matching [9]. While the majority of the existing methods were developed for the task of matching database records, modern approaches focus mostly on graph-based data representations extended by additional schema information. We discuss the problem of object reconciliation using the notion of instance matching. This allows us to describe it within the well-established ontology matching framework [7]. Ontology matching is the process of detecting links between entities in different ontologies. These links are annotated by a confidence value and a label describing the type of link. Such a link is referred to as a *correspondence* and a set of such correspondences is referred to as an *alignment*.

Definition 1 (Correspondence and Alignment). *Given ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, let q be a function that defines sets of matchable entities $q(\mathcal{O}_1)$ and $q(\mathcal{O}_1)$. A correspondence between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a four tuple $\langle e_1, e_2, r, n\rangle$ such that $e_1 \in q(\mathcal{O}_1)$ and $e_2 \in q(\mathcal{O}_2)$, r is a semantic relation and n is a confidence value. An alignment $\mathcal{M}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a set of correspondences between $\mathcal{O}_1$ and $\mathcal{O}_2$.*

The generic form of Definition 1 captures a wide range of correspondences by varying what is admissible as matchable element, semantic relation, and confidence value. A fundamental distinction between different matching tasks is determined by the restriction q on the set of matchable entities. On the one hand we might be interested in links between terminological entities (concepts and properties) and on the other hand we might want to find links between instances. In the following we refer to an alignment that contains correspondences of the former type as *terminological alignment* and to an alignment that contains correspondences of the latter type as *instance alignment*. Terminological alignments relate the T-Boxes of $\mathcal{O}_1$ and $\mathcal{O}_2$ by providing equivalence or subsumption links between concepts and properties. Since instance matching is the task of detecting pairs of instances that refer to the same real world object [9], the semantic relation expressed by an instance correspondence is that of identity. The confidence value of a correspondence quantifies the degree of trust in the correctness of the statement. If a correspondence is automatically generated by a matching system this value will be computed by aggregating scores from different sources of evidence. The commonly applied methods for object reconciliation include structure-based strategies as well as strategies to compute and aggregate value similarities. Under the notion of instance matching, similarities between instance labels and datatype properties are mostly used to compute confidence values for instance correspondences. Examples of this are realized in the systems RiMOM [23] and OKKAM [18]. Additional refinements are related to a distinction between different types of properties. The developers of RiMOM manually distinguish between *necessary* and *sufficient* datatype properties. The FBEM algorithm of the OKKAM project assigns higher weights to certain properties like names and IDs. In both cases, the employed methods focus on appropriate techniques to interpret and aggregate similarity scores based on a comparison of datatype property values. Another important source of evidence is the knowledge encoded in the T-Box. RiMOM, for example, first generates a terminological alignment between the T-Boxes $\mathcal{T}_1$ and $\mathcal{T}_2$ describing the A-Boxes $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. This alignment is then used as a filter and only correspondences that link instances of equivalent concepts are considered valid [23].

In this paper we are concerned with the scenario where both A-Boxes are described in terms of the same T-Box. An object reconciliation method applicable to this setting is also proposed in [15] where the authors combine logical with numerical methods. For logical reasons it is in some cases possible to preclude that two instances refer to the same object while in other cases the acceptance of one correspondence directly entails the acceptance of another. The authors extend this approach by modeling some of these dependencies into a similarity propagation framework. However, their approach requires a rich schema and assumes that properties are defined to be functional and/or inverse functional. Hence, the approach cannot be used effectively to exploit type information based on a concept hierarchy and is therefore not applicable in many web of data scenarios. In contrast, our approach does not rely on specific types of axioms or a set of predefined rules but on a well defined semantic similarity measure. A num-

ber of different approaches to quantify the degree of similarity between concept descriptions and ontologies have been proposed [2]. In particular, our approach is based on the measure proposed by Stuckenschmidt [19]. This measure has originally been designed to quantify the similarity between two ontologies that describe the same set of objects. We apply a modified variant of this measure to evaluate the similarity of two A-Boxes described in terms of the same T-Box. Furthermore, our method factors in a-priori confidence values that quantify the degree of trust one has in the correctness of the object correspondences based on lexical properties. The resulting similarity measure is used to determine an instance alignment that induces the highest agreement of object assertions in $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to $\mathcal{T}$.

3 A Similarity Measure for Instance Matching

In [19] Stuckenschmidt introduces a measure that quantifies the similarity of two A-Boxes described in terms of the same T-Box. A brief description is given in Section 3.1. In Section 3.2 we propose a modification of this measure that factors in a-priori confidence values. We argue that the underlying idea of the measure can be used to appropriately incorporate T-Box information during the matching process. Additionally, we explain and motivate our approach by means of an example. In the following, we will use $\langle a, b\rangle$ to refer to an instance correspondence $\langle a, b, =, n\rangle$ and the a-priori similarity $\sigma(a, b)$ to refer to the confidence value n.

3.1 Measuring A-Box Similarity

Stuckenschmidt's similarity measure is based on the notion of a *valid* instance alignment. Given an instance alignment $\mathcal{M}$ between $\mathcal{A}_1$ and $\mathcal{A}_2$, suppose that we merge both $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{T}$, and $\mathcal{M}$ into a single ontology $\mathcal{O}$. Due to some mismatches in $\mathcal{M}$ it might happen that $\mathcal{O}$ becomes inconsistent. Obviously, we want to avoid alignments that lead to inconsistencies. The following definition formally introduces the notion of a valid alignment.

Definition 2 (Valid Alignment). *Let $\mathcal{M}$ be an instance alignment between A-Boxes $\mathcal{A}_1$ and $\mathcal{A}_2$ both described in terms of T-Box $\mathcal{T}$. $\mathcal{M}$ is valid with respect to $\mathcal{T}$ if and only if for all concepts C and all properties P defined in $\mathcal{T}$ as well as for all correspondences $\langle a, b\rangle, \langle a', b'\rangle \in \mathcal{M}$ we have*

$$\begin{aligned}
\mathcal{T} \cup \mathcal{A}_1 \models C(a) \;&\Rightarrow\; \mathcal{T} \cup \mathcal{A}_2 \not\models \neg C(b) \\
\mathcal{T} \cup \mathcal{A}_2 \models C(b) \;&\Rightarrow\; \mathcal{T} \cup \mathcal{A}_1 \not\models \neg C(a) \\
\mathcal{T} \cup \mathcal{A}_1 \models P(a, a') \;&\Rightarrow\; \mathcal{T} \cup \mathcal{A}_2 \not\models \neg P(b, b') \\
\mathcal{T} \cup \mathcal{A}_2 \models P(b, b') \;&\Rightarrow\; \mathcal{T} \cup \mathcal{A}_1 \not\models \neg P(a, a')
\end{aligned}$$

Under the assumption that two different URI references in the same A-Box denote two distinct instances, a *valid* alignment will not lead to inconsistencies in the merged ontology. We now introduce the notion of a *functional one-to-one*

alignment between A-Boxes. $\mathcal{M}$ is a functional one-to-one alignment if and only if for all pairs of correspondences $\langle a, b\rangle \neq \langle a', b'\rangle \in \mathcal{M}$ we have $a \neq a'$ and $b \neq b'$. Based on the notion of a valid functional one-to-one alignment one can count, for each possible alignment $\mathcal{M}$, the number of assertions identical in $\mathcal{A}_1$ and $\mathcal{A}_2$, where instance equivalence is determined by the alignment $\mathcal{M}$. We will call this value the *overlap* of two A-Boxes $\mathcal{A}_1$ and $\mathcal{A}_2$ induced by $\mathcal{M}$.

Definition 3 (Overlap). *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be A-Boxes described in terms of T-Box $\mathcal{T}$. Furthermore, let $\mathcal{M}$ be a functional one-to-one*[1] *instance alignment between $\mathcal{A}_1$ and $\mathcal{A}_2$ that is valid with respect to $\mathcal{T}$. The overlap of $\mathcal{A}_1$ and $\mathcal{A}_2$ induced by $\mathcal{M}$ with respect to $\mathcal{T}$ is defined as*

$$
\begin{aligned}
&\mathit{overlap}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{M}) := \\
&\quad |\, \{C(a) \quad\; | \; \mathcal{T} \cup \mathcal{A}_1 \models C(a) \;\wedge\; \mathcal{T} \cup \mathcal{A}_2 \models C(b) \;\wedge\; \langle a, b\rangle \in \mathcal{M}\} \cup \\
&\quad\; \{\neg C(a) \quad | \; \mathcal{T} \cup \mathcal{A}_1 \models \neg C(a) \;\wedge\; \mathcal{T} \cup \mathcal{A}_2 \models \neg C(b) \;\wedge\; \langle a, b\rangle \in \mathcal{M}\} \cup \\
&\quad\; \big\{P(a, a') \;\; | \; \mathcal{T} \cup \mathcal{A}_1 \models P(a, a') \;\wedge\; \mathcal{T} \cup \mathcal{A}_2 \models P(b, b') \;\wedge\; \langle a, b\rangle, \langle a', b'\rangle \in \mathcal{M}\big\} \cup \\
&\quad\; \big\{\neg P(a, a') \,|\, \mathcal{T} \cup \mathcal{A}_1 \models \neg P(a, a') \;\wedge\; \mathcal{T} \cup \mathcal{A}_2 \models \neg P(b, b') \;\wedge\; \langle a, b\rangle, \langle a', b'\rangle \in \mathcal{M}\big\} \,|
\end{aligned}
$$

Based on this, it is possible to define the A-Box similarity between $\mathcal{A}_1$ and $\mathcal{A}_2$ as the maximal possible overlap of $\mathcal{A}_1$ and $\mathcal{A}_2$. In order to find this value, we have to consider the set of all possible valid functional one-to-one alignments $\mathbb{M}$ between $\mathcal{A}_1$ and $\mathcal{A}_2$. Notice that the overlap is not only determined by $\mathcal{M}$ but also by the size of $\mathcal{A}_1$, $\mathcal{A}_2$ (number of instances), and $\mathcal{T}$ (number of concepts and properties). Thus, we have to use a normalizing denominator. The resulting similarity measure quantifies the degree of similarity as a value in the interval $[0, 1]$.

Definition 4 (A-Box Similarity). *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be A-Boxes described in terms of T-Box $\mathcal{T}$. Furthermore, let $\mathbb{M}$ be the set of all functional one-to-one instance alignments between $\mathcal{A}_1$ and $\mathcal{A}_2$ that are valid with respect to $\mathcal{T}$. The A-Box similarity between $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to $\mathcal{T}$ is defined as*

$$
\mathit{sim}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) := \max_{\mathcal{M} \in \mathbb{M}} \frac{2 * \mathit{overlap}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{M})}{\mathit{overlap}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_1, \mathcal{I}_{\mathcal{A}_1}) + \mathit{overlap}_{\mathcal{T}}(\mathcal{A}_2, \mathcal{A}_2, \mathcal{I}_{\mathcal{A}_2})}
$$

where $\mathcal{I}_{\mathcal{A}}$ refers to the identity alignment that maps every instance described in an A-Box $\mathcal{A}$ on itself.

Notice that this similarity measure fulfills the properties of a conceptual similarity measure as defined by Amato et al. [5]. In particular, we have $0 \leq \mathit{sim}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) \leq 1$, $\mathit{sim}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2) = \mathit{sim}_{\mathcal{T}}(\mathcal{A}_2, \mathcal{A}_1)$, and $\mathit{sim}_{\mathcal{T}}(\mathcal{A}, \mathcal{A}) = 1$.

[1] The approach is not limited to functional one-to-one alignments but can also generate m-to-n alignments. To simplify the exposition of the framework, however, we chose to describe it with respect to functional one-to-one alignments.

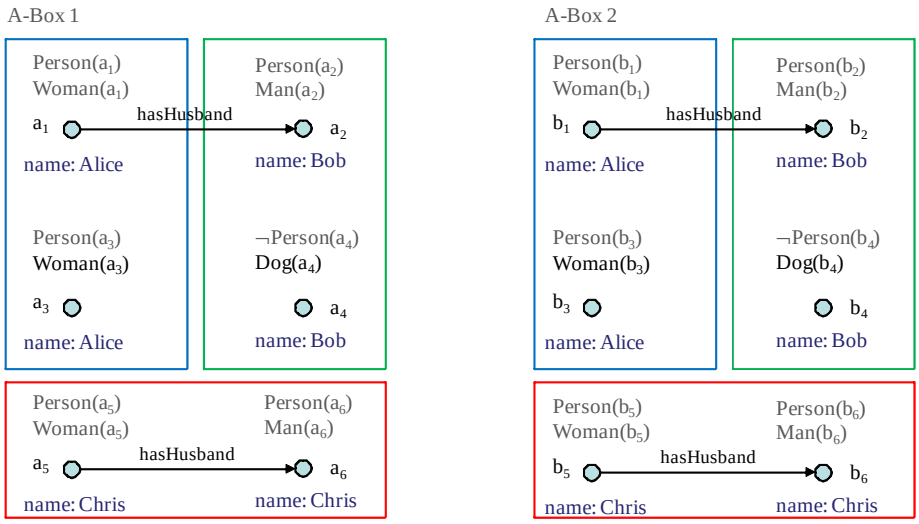

Fig. 1. Motivating example.

3.2 Exploiting A-Box Similarity

In this section, we leverage the A-Box similarity from Definition 4 for the task of object reconciliation. Furthermore, we demonstrate the advantage of our method over those approaches using only lexical confidence values. Therefore, we introduce a small motivating example. Suppose that the shared T-Box $\mathcal{T}$ is defined as follows.

$$\begin{aligned} \exists hasHusband &\sqsubseteq Woman \\ \exists hasHusband^{-} &\sqsubseteq Man \\ Dog &\sqsubseteq \neg Person \\ Person &\equiv Woman \sqcup Man \end{aligned}$$

Let us assume we have six individuals $a_1, ..., a_6$ and $b_1, ..., b_6$ in each A-Box. Furthermore, let us assume that the following concept and object property assertions are explicitly specified in $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.

$$\begin{aligned} hasHusband(a_1, a_2) \quad & hasHusband(b_1, b_2) \\ hasHusband(a_5, a_6) \quad & hasHusband(b_5, b_6) \\ Woman(a_3) \quad & Woman(b_3) \\ Dog(a_4) \quad & Dog(b_4) \end{aligned}$$

Figure 1 provides an illustration of this example. Stated assertions are depicted in black, while gray-colored assertions can be inferred from the given ones with respect to the T-Box $\mathcal{T}$. To simplify the notation, we base the a-priori confidence value $\sigma(a, b)$ of a correspondence $\langle a, b \rangle$ on the identity of the *name* datatype-property value by setting the lexical similarity to 1 if the strings of the *name*

attribute are identical and to 0 otherwise. Now, if we computed the individual alignment between $\mathcal{A}_1$ and $\mathcal{A}_2$ by maximizing only the given lexical similarity, we would not be able to differentiate between the pairs of individuals with name *Alice* (blue squares including a_1, a_3 and b_1, b_3 in Figure 1), with name *Bob* (green squares including a_2, a_4 and b_2, b_4 in Figure 1), and with name *Chris* (red squares including a_5, a_6 and b_5, b_6 in Figure 1), respectively. Consequently, there would only be a probability of $\frac{1}{8}$ of choosing the correct alignment. In light of this, we introduce Definition 5 which extends the notion of A-Box overlap by incorporating the (a-priori) lexical confidence values as coefficients.

Definition 5 (Weighted Overlap). *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be A-Boxes both described in terms of T-Box $\mathcal{T}$. Furthermore, let $\mathcal{M}$ be a functional one-to-one instance alignment between $\mathcal{A}_1$ and $\mathcal{A}_2$ that is valid with respect to $\mathcal{T}$ and with a-priori confidence values given by σ. The weighted overlap between $\mathcal{A}_1$ and $\mathcal{A}_2$ induced by $\mathcal{M}$ with respect to $\mathcal{T}$ is defined as*

$$overlap^{w}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{M}) :=$$

$$\sum_{\langle a,b\rangle \in \mathcal{M}} \sum_{\substack{C \in \mathcal{T}: \\ \mathcal{T} \cup \mathcal{A}_1 \models C(a) \ \wedge \\ \mathcal{T} \cup \mathcal{A}_2 \models C(b)}} \sigma(a,b) + \sum_{\langle a,b\rangle, \langle a',b'\rangle \in \mathcal{M}} \sum_{\substack{P \in \mathcal{T}: \\ \mathcal{T} \cup \mathcal{A}_1 \models P(a,a') \ \wedge \\ \mathcal{T} \cup \mathcal{A}_2 \models P(b,b')}} \frac{\sigma(a,b) + \sigma(a',b')}{2} +$$

$$\sum_{\langle a,b\rangle \in \mathcal{M}} \sum_{\substack{C \in \mathcal{T}: \\ \mathcal{T} \cup \mathcal{A}_1 \models \neg C(a) \ \wedge \\ \mathcal{T} \cup \mathcal{A}_2 \models \neg C(b)}} \sigma(a,b) + \sum_{\langle a,b\rangle, \langle a',b'\rangle \in \mathcal{M}} \sum_{\substack{P \in \mathcal{T}: \\ \mathcal{T} \cup \mathcal{A}_1 \models \neg P(a,a') \ \wedge \\ \mathcal{T} \cup \mathcal{A}_2 \models \neg P(b,b')}} \frac{\sigma(a,b) + \sigma(a',b')}{2}.$$

The main difference between Definition 5 and Definition 3 is the weighing of the overlap of every (negated) concept and object property assertion with the a-priori similarity σ. We revisit our example to verify the ability of Definition 5 to leverage positive and negative concept and object property assertions and to improve the quality of the alignment. In order to show the improvements, we compare the weighted overlap score of the different possibilities to align the individuals named *Bob*, *Alice*, and *Chris*. With respect to the individuals named *Chris* there are two possible alignments $\{\langle a_5, b_5\rangle, \langle a_6, b_6\rangle\}$ and $\{\langle a_5, b_6\rangle, \langle a_6, b_5\rangle\}$. Both alternatives link individuals that belong to the same concept $Person$ and, therefore, both add a score of two to the weighted overlap. In addition, the partial alignment $\{\langle a_5, b_5\rangle, \langle a_6, b_6\rangle\}$ links the instances having the concepts $Woman$ and Man in common. Consequently, this combination adds an additional score of two to the weighted A-Box similarity. As a result, our approach will make the partial alignment $\{\langle a_5, b_5\rangle, \langle a_6, b_6\rangle\}$ part of the optimal valid one-to-one alignment due to the greater overlap of concept-assertions.

In case of the individuals named *Alice* the two possible partial alignments $\{\langle a_1, b_1\rangle, \langle a_3, b_3\rangle\}$ and $\{\langle a_1, b_3\rangle, \langle a_3, b_1\rangle\}$ exist. All individuals named *Alice* belong to the concepts $Woman$ and $Person$. This means that concept assertions are not sufficient to distinguish between these alignments. However, the existing object property assertions $hasHusband(a_1, a_2)$ and $hasHusband(b_1, b_2)$ increase

the weighted similarity only for the alignment containing $\{\langle a_1, b_1\rangle, \langle a_3, b_3\rangle\}$. Accordingly, our method will make the partial alignment $\{\langle a_1, b_1\rangle, \langle a_3, b_3\rangle\}$ part of the optimal valid one-to-one alignment because the approach also takes object property assertions into account.

Finally, for the individuals named *Bob* the partial alignments under consideration are $\{\langle a_2, b_2\rangle, \langle a_4, b_4\rangle\}$ and $\{\langle a_2, b_4\rangle, \langle a_4, b_2\rangle\}$. Due to the disjointness axiom specified in $\mathcal{T}$ and the existing assertions one can infer the negative concept assertions $\neg Person(a_4)$ and $\neg Person(b_4)$. Hence, according to Definition 2, an alignment containing both $\langle a_2, b_4\rangle$ and $\langle a_4, b_2\rangle$ is not valid. Therefore, our method will make the partial alignment $\{\langle a_2, b_2\rangle, \langle a_4, b_4\rangle\}$ part of the optimal valid one-to-one alignment. This illustrates how our approach also factors in negative concept and object property assertions.

4 Optimal and Approximate Algorithms for Computing the Maximal Weighted A-Box Similarity

We now turn to the problem of devising algorithms that compute the previously defined (weighted) similarity measure between A-Boxes. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two A-Boxes both described in terms of a T-Box $\mathcal{T}$. It follows from Definition 4 and Definition 5 that, in order to compute the alignment that maximizes the weighted A-Box similarity, we have to determine

$$\underset{\mathcal{M}\in\mathbb{M}}{\operatorname{argmax}}\, overlap^{w}_{\mathcal{T}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{M})$$

with $\mathbb{M}$ the set of all functional one-to-one instance alignments that are valid with respect to $\mathcal{T}$. Notice that we can ignore the normalization denominator from Definition 4 since we are not directly interested in the maximal weighted A-Box similarity but rather the alignment that maximizes it. The problem of finding this alignment is computationally challenging due to its combinatorial complexity. It is essentially equivalent to the inexact multi-labeled graph matching problem, except that the validity requirement from Definition 2 can potentially lead to additional constraints on the set of possible alignments. As the inexact multi-labeled graph matching problem is NP-complete because it generalizes the well-known subgraph isomorphism problem [14], it can be shown that finding the alignment that maximizes the weighted A-Box similarity is also an NP-hard problem[2]. Nevertheless, we are able to provide efficient algorithms by (a) transforming the problem into an integer linear programming problem [16], and by (b) applying the approximate multi-labeled graph matching algorithm of Cour et al. [4] to the problem. We discuss the details of these two approaches in the remainder of this section.

[2] To prove the NP-hardness one can construct, for every instance of the multi-labeled graph matching problem, two A-Boxes $\mathcal{A}_1$ and $\mathcal{A}_2$ such that the alignment that maximizes the weighted A-Box similarity between $\mathcal{A}_1$ and $\mathcal{A}_2$ is also the solution to the corresponding inexact multi-labeled graph matching problem. We omit the details as the proof is beyond the scope of the paper.

4.1 Integer Linear Programming

Integer linear programming (ILP) can be defined as the problem of optimizing a linear objective function over a finite number of integer variables, subject to a set of linear equalities and inequalities over these variables. It is a problem that mainly occurs in the field of operations research [20]. From a mathematical perspective, it can be defined as the problem of finding a point on a polyhedron, determined by the given linear (in-)equalities, at which the linear objective function attains its minimum or maximum [16]. The problem of finding the alignment that maximizes the weighted similarity of two A-Boxes can be transformed to a integer linear programming problem as follows.

Variables: Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two A-Boxes described in terms of a T-Box $\mathcal{T}$ and let a_i, $1 \leq i \leq n$ and b_j, $1 \leq j \leq m$, denote the individuals in $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. We will denote the set of variables of the ILP with V. Now, for every $1 \leq i \leq n$ and $1 \leq j \leq m$, we add the variable $x_{\langle i,j \rangle}$ to the set V if there exists at least one concept[3] $C \in \mathcal{T}$ such that either $\mathcal{T} \cup \mathcal{A}_1 \models C(a_i)$ and $\mathcal{T} \cup \mathcal{A}_2 \models C(b_j)$ *or* $\mathcal{T} \cup \mathcal{A}_1 \models \neg C(a_i)$ and $\mathcal{T} \cup \mathcal{A}_2 \models \neg C(b_j)$. In addition, for every $1 \leq i,k \leq n$ and $1 \leq j,l \leq m$, we add the variables $x_{\langle i,j \rangle}$, $x_{\langle k,l \rangle}$, and $s_{\langle i,j \rangle,\langle k,l \rangle}$ to the set V if there exists at least one object property $P \in \mathcal{T}$ with either $\mathcal{T} \cup \mathcal{A}_1 \models P(a_i,a_k)$ and $\mathcal{T} \cup \mathcal{A}_2 \models P(b_j,b_l)$ *or* $\mathcal{T} \cup \mathcal{A}_1 \models \neg P(a_i,a_k)$ and $\mathcal{T} \cup \mathcal{A}_2 \models \neg P(b_j,b_l)$. We will require all variables in V to be binary, that is, they can take on the values 0 and 1, respectively. Note that variable $x_{\langle i,j \rangle}$ represents the correspondence $\langle a_i, b_j \rangle$, that is, $x_{\langle i,j \rangle}$ will be 1 in the solution of the ILP if and only if the correspondence $\langle a_i, b_j \rangle$ is part of the alignment that maximizes the weighted A-Box similarity. Furthermore, the variable $s_{\langle i,j \rangle,\langle k,l \rangle}$ represents the correspondences $\langle a_i, b_j \rangle$ and $\langle a_k, b_l \rangle$, that is, $s_{\langle i,j \rangle,\langle k,l \rangle}$ will be 1 in the solution of the ILP if and only if both correspondences $\langle a_i, b_j \rangle$ and $\langle a_k, b_l \rangle$ are part of the alignment that maximizes the weighted A-Box similarity.

Objective Function: We will now define the coefficient for each of the variables in V. For every $x_{\langle i,j \rangle} \in V$ we set the coefficient $c_{\langle i,j \rangle}$ to be the product of the a-priori similarity of the individuals a_i and b_j and the number of (negated) concepts in $\mathcal{T}$ of which both a_i and b_j are instances:

$$c_{\langle i,j \rangle} := \sigma(a_i,b_j) * |\{C \mid \mathcal{T} \cup \mathcal{A}_1 \models C(a_i) \wedge \mathcal{T} \cup \mathcal{A}_2 \models C(b_j)\} \cup \{C \mid \mathcal{T} \cup \mathcal{A}_1 \models \neg C(a_i) \wedge \mathcal{T} \cup \mathcal{A}_2 \models \neg C(b_j)\}|$$

Similarly, for every $s_{\langle i,j \rangle,\langle k,l \rangle} \in V$ we set the coefficient $d_{\langle i,j \rangle,\langle k,l \rangle}$ to be the product of the mean of the a-priori similarities between the individuals a_i, b_j and a_k,b_l, respectively, and the number of (negated) object properties in $\mathcal{T}$ of which both pairs $\langle a_i, a_k \rangle$ and $\langle b_j, b_l \rangle$ are instances:

$$d_{\langle i,j \rangle,\langle k,l \rangle} := (\sigma(a_i,b_j) + \sigma(a_k,b_l))/2 * |\{P \mid \mathcal{T} \cup \mathcal{A}_1 \models P(a_i,a_k) \wedge \mathcal{T} \cup \mathcal{A}_2 \models P(b_j,b_l)\} \cup \{P \mid \mathcal{T} \cup \mathcal{A}_1 \models \neg P(a_i,a_k) \wedge \mathcal{T} \cup \mathcal{A}_2 \models \neg P(b_j,b_l)\}|$$

[3] We do *not* consider the top concept *thing* in the formulation of the ILP.

Finally, we can define the objective of the ILP as

$$\text{Maximize:} \quad \sum_{x_{\langle i,j\rangle} \in V} c_{\langle i,j\rangle} x_{\langle i,j\rangle} + \sum_{s_{\langle i,j\rangle,\langle k,l\rangle} \in V} d_{\langle i,j\rangle,\langle k,l\rangle} s_{\langle i,j\rangle,\langle k,l\rangle}$$

Linear Constraints: In addition to the variables and the objective function we also need to introduce several linear constraints to ensure that every feasible solution of the ILP corresponds to a valid functional one-to-one alignment between the A-Boxes $\mathcal{A}_1$ and $\mathcal{A}_2$. First, we enforce that every solution of the ILP corresponds to an alignment that is both (a) one-to-one and (b) functional by introducing the following sets of constraints:

$$\text{(a) } \forall j : \sum_{x_{\langle i,j\rangle} \in V} x_{\langle i,j\rangle} \leq 1 \text{ and (b) } \forall i : \sum_{x_{\langle i,j\rangle} \in V} x_{\langle i,j\rangle} \leq 1.$$

Furthermore, for any solution of the ILP, every variable $s_{\langle i,j\rangle,\langle k,l\rangle} \in V$ will be set to 1 if and only if the two corresponding variables $x_{\langle i,j\rangle}$ and $x_{\langle k,l\rangle}$ are also both set to 1. This can be modeled with a conjunction of the following three constraints:

$$s_{\langle i,j\rangle,\langle k,l\rangle} - x_{\langle i,j\rangle} \leq 0;\ s_{\langle i,j\rangle,\langle k,l\rangle} - x_{\langle k,l\rangle} \leq 0;\ \text{and } x_{\langle i,j\rangle} + x_{\langle k,l\rangle} - s_{\langle i,j\rangle,\langle k,l\rangle} \leq 1.$$

Finally, the validity requirement introduced in Definition 2 has to be enforced. For every variable $x_{\langle i,j\rangle} \in V$ we add the linear constraint $x_{\langle i,j\rangle} \leq 0$ if there exists at least one concept $C \in \mathcal{T}$ with $\mathcal{A}_1 \cup \mathcal{T} \models C(a_i)$ and $\mathcal{A}_2 \cup \mathcal{T} \models \neg C(b_j)$ *or* $\mathcal{A}_1 \cup \mathcal{T} \models \neg C(a_i)$ and $\mathcal{A}_2 \cup \mathcal{T} \models C(b_j)$. In addition, for every pair of variables $x_{\langle i,j\rangle} \in V$ and $x_{\langle k,l\rangle} \in V$ we add the linear constraint $x_{\langle i,j\rangle} + x_{\langle k,l\rangle} \leq 1$ to the ILP if there exists at least one object property $P \in \mathcal{T}$ with $\mathcal{A}_1 \cup \mathcal{T} \models P(a_i, a_k)$ and $\mathcal{A}_2 \cup \mathcal{T} \models \neg P(b_j, b_l)$ *or* $\mathcal{A}_1 \cup \mathcal{T} \models \neg P(a_i, a_k)$ and $\mathcal{A}_2 \cup \mathcal{T} \models P(b_j, b_l)$. Note that an additional advantage of the method is the possibility to add *known correct* correspondences to the formulation of the ILP.

The proof of the following theorem is omitted due to space constraints.

Theorem 1. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two A-Boxes described in terms of a T-Box $\mathcal{T}$. Furthermore, let ILP be the integer linear program constructed from $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{T}$ according to the previous steps. Then every set of variables comprising a solution of ILP correspond to an alignment that maximizes the weighted A-Box similarity between $\mathcal{A}_1$ and $\mathcal{A}_2$.*

4.2 Approximate Algorithm

In the experimental section, we will verify empirically that the transformation to an integer linear program can be efficiently solved for small to medium sized ontologies. However, due to the inherent computational complexity of the problem, the method will not scale to ontologies with large numbers of instances. Therefore, we will additionally apply an inexact graph matching algorithm [4] to approximate the alignment that maximizes the weighted A-Box similarity. This algorithm was originally developed for graph matching problems occurring

in the areas of computer vision and machine learning. It solves a continuous relaxation of an integer quadratic programming formulation of the inexact graph matching problem and is closely related to the spectral matching formulation of [12]. The construction of the quadratic formulation is similar to the previous construction of the ILP. We refer the interested reader to these articles for a more detailed description of the algorithm.

5 Experimental Evaluation

Now that we have introduced our framework for instance matching we will present empirical evidence for the utility of the method on real-world object reconciliation problems. We conducted the experiments with the following questions in mind:

- To which degree can we improve standard instance matching approaches which are mostly based on lexical similarities between datatype properties?
- How efficient is our approach with respect to runtime?
- How well does the approximate graph matching algorithm perform compared to the ILP approach?

Before we present the results of the experiments, we describe the datasets we used for our experiments as well as the baseline algorithms against which we compare our methods.

5.1 Dataset and Experimental Set-up

We used the IIMB benchmark dataset[4] for the experiments. The benchmark was developed by Ferrara et al. [9] and provides a set of realistic object reconciliation problems with each of the A-Boxes containing about 300 individuals. The individuals are specific movies, actors, and directors. The T-Box is that of a typical light-weight ontology with 5 concepts, 13 datatype, and 5 object properties. There is one reference dataset with the original T-Box and A-Box and 70 different transformations which can be roughly divided into the following four categories:

Values Transformations (VT): Typographical errors are simulated and other lexical modifications like changing the word order are applied to datatype property values.

Structural Transformations (ST): The focus of these transformations is on the modification of the datatype properties themselves. They include value deletions, depth modifications, and value separations.

Combination of VT and ST (VT & ST): The combination of the previous two types of transformations.

Logical Transformations (LT): Instances are moved to different classes. These classes may be disjoint, explicit/implicit subclasses, or entirely new concepts in the T-Box.

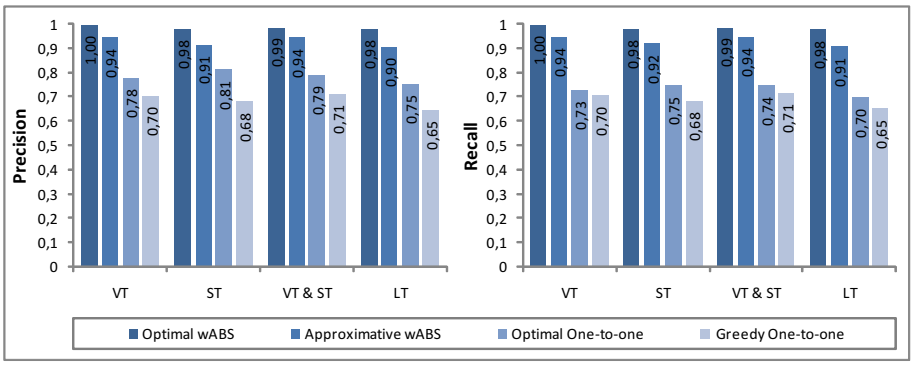

Fig. 2. Precision and recall for the four different methods.

For our experiments we implemented a simple measure to compute the lexical a-priori similarity σ. For each pair of individuals a and b it considers the datatype properties $p_1, ..., p_n$ that are used to describe both a and b. The a-priori similarity is then defined as the average similarity of these datatype property values: $\sigma(a,b) = \frac{1}{n}\sum_{i=1}^{n} sim(p_i(a), p_i(b))$. We set $\sigma(a,b)$ to zero for all pairs of individuals that have no datatype properties in common. To quantify the lexical similarity sim we used the SoftTFIDF string matching approach introduced by Cohen et al. [3] without modifications. Note again that the a-priori similarity σ can be replaced by any other measure that estimates the lexical similarity of individuals. For a survey on existing methods in the context of record linkage we refer the reader to Elmagarmid [6]. Once an appropriate similarity measure σ is chosen, most state-of-the-art approaches use one of the following two methods to generate functional one-to-one alignments. The first method selects, from the set of possible correspondences, the one correspondence $\langle a, b\rangle$ with highest confidence $\sigma(a,b)$ and removes all correspondences containing either a or b from the set of possible correspondences. This procedure is repeated until a functional one-to-one alignment is generated. We refer to this method as *greedy one-to-one*. The second method (denoted as *optimal one-to-one*) computes the functional one-to-one alignment that maximizes the sum of the a-priori confidence values. We implemented both methods and used the results obtained as baselines in our experiments.

We denote the optimal algorithm as *optimal wABS* (weighted A-Box Similarity) and the approximate graph matching algorithm as *approximate wABS*[5]. We used the mixed integer programming algorithm SCIP[6] to solve the ILP formulation of the *optimal wABS* algorithm. According to standard benchmarks for mixed integer linear programming algorithms[7], SCIP is one of the fastest non-commercial solvers. However, there are commercial solvers available which

[4] `http://islab.dico.unimi.it/content/iimb2009/`

[5] The Matlab implementation is available at `http://www.seas.upenn.edu/~timothee/software/graph_matching/graph_matching.html`

[6] `http://scip.zib.de`

[7] `http://plato.asu.edu/ftp/milpf.html`

	Optimal wABS			Approximate wABS		
	overall	**load and reason**	**execute algorithm**	**overall**	**load and reason**	**execute algorithm**
Mean	4774.7	119,5	4728,4	146.2	121,5	22,5
Median	3221.5	123,2	3061,5	148.0	122,7	24,8
St. Dev.	4979.3	13,7	5158,1	19.1	10,6	5,2

Table 1. Average execution times (in seconds) for the two different methods.

are several times faster than SCIP according to the benchmarks. Also, since the ILP formulation is independent of the particular solving method, progress in mixed integer linear programming will directly translate to shorter runtimes of the *optimal wABS* algorithm. The logical reasoning necessary to prepare the input for the *optimal and approximate wABS* algorithms was carried out using the reasoner Pellet [17]. All experiments were run on a desktop PC with an AMD Athlon dual core 6000+ 3.01 GHz processor and 3 GB RAM.

5.2 Results

We first evaluated the performance of the two baseline algorithms by comparing them to existing OAEI 2009 results of state-of-the-art matching systems. The *greedy* and *optimal one-to-one* algorithms based on the rather simple average lexical similarity achieved higher precision and recall values than 2 of the 6 state-of-the-art matchers. Hence, our baseline algorithms are comparable to the performance of existing matching algorithms. We then ran all four algorithms on the different modifications included in the IIMB dataset[8]. Figure 2 depicts the average recall and average precision values for the four categories. The results show a significant increase of precision and recall for the two wABS methods compared to the two baseline one-to-one algorithms. The *approximative wABS* algorithm has a precision and recall of 0.92 and 0.93, respectively, while the *optimal wABS* algorithm reaches a precision and recall of 0.99. Comparing this to the precision and recall of the *optimal one-to-one* algorithm of 0.78 and 0.73, respectively, we have a solid improvement between 18% and 36%. These results verify that leveraging T-Box information significantly improves the accuracy of alignments. They also show the trade-off between runtime and accuracy. The *approximative wABS* algorithm has lower precision and recall than the optimal method but is about 30 times faster. Table 1 depicts the execution times (*including* reasoning and preprocessing) of these two algorithms. While the *optimal wABS* algorithm needs an average of 1.3 hours to compute the alignment, the *approximate wABS* algorithm needs only about 2 minutes. The high standard deviation of the *optimal wABS* method speaks to the computational complexity of the problem. In most cases the ILP solver finds the optimal solution relatively fast but due to the hardness of the problem there are naturally some hard cases which increase the average runtime of the algorithm.

[8] We had to omit 5 of the 70 variations (19, 21, 37, 39, and 40) since these cases involved different T-Boxes.

Overall, the experiments demonstrate that using the weighted A-Box similarity improves the instance alignments substantially. Since instance matching is usually not time-critical the *optimal wABS* algorithm is applicable to small to medium sized ontologies. The *approximate wABS* algorithm has the potential to also scale to larger ontologies with only a modest loss of precision and recall. The complete experimental results and the implementations are available at `http://webrum.uni-mannheim.de/math/lski/matching/rec/`.

6 Discussion and Future Work

We proposed a framework for object reconciliation based on a semantic similarity measure between A-Boxes. The framework allows one to combine lexical a-priori similarities between instances with the terminological knowledge encoded in the ontology. We argued that most state-of-the-art approaches for instance matching focus solely on ways to compute lexical similarities. These approaches are sometimes extended by a structural validation technique where class membership is used as a matching filter. However, even though useful in some scenarios, these methods are neither based on a well defined theoretical framework nor generally applicable without adjustment. Contrary to this, our approach is grounded in a coherent theory and incorporates terminological knowledge during the matching process. Our experiments show that the resulting method is flexible enough to cope with difficult matching problems for which lexical similarity alone is not sufficient to ensure high-quality instance alignments.

Currently, our approach is restricted to generate alignments between A-Boxes described in terms of the same T-Box. In some cases this requirement is unrealistic. In such a situation it might make sense to merge the two T-Boxes prior to the instance matching process. Especially in cases where we have large A-Boxes described with relatively small T-Boxes, the benefits demonstrated by our experiments legitimate the required manual effort. In addition to this, our framework can be extended to generate both instance *and* terminological alignments at the same time. This extension requires to model instance and terminological correspondences in the same way. Instead of interpreting the axioms of the shared T-Box as hard constraints, we have to interpret both types of correspondences as soft constraints. This way we benefit from an automatically generated, uncertain terminological alignment while avoiding the risk of rejecting correct instance correspondences. In this setting, both types of correspondences are in contest with each other. The solution to the corresponding optimization problem leads to both an instance alignment and a terminological alignment. Further research will show whether this approach will provide a general framework for ontology matching that unifies instance and schema matching in an appropriate way.

Acknowledgement: We thank Alfino Ferrara for providing us the IIMB benchmark and for the initiative at `http://www.instancematching.org/`.

References

1. Indrajit Bhattacharya and Lise Getoor. Entity resolution in graphs. In *Mining Graph Data*. Wiley & Sons, 2006.
2. Alex Borgida, Thomas J. Walsh, and Haym Hirsh. Towards measuring similarity in description logics. In *Proceedings of DL*, 2005.
3. W.W. Cohen, P. Ravikumar, and S.E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, 2003.
4. T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *Advances in Neural Information Processing Systems 19*, 2007.
5. C. D'Amato, S. Staab, and N. Fanizzi. On the influence of description logics ontologies on conceptual similarity. In *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, 2008.
6. A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering*, 19(1):1, 2007.
7. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, 2007.
8. Ivan Fellegi and Alan Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
9. A. Ferrara, D. Lorusso, S. Montanelli, and G. Varese. Towards a Benchmark for Instance Matching. In *The 7th International Semantic Web Conference*, 2008.
10. O. Hassanzadeh, L. Lim, A. Kementsietsidis, and M. Wang. A declarative framework for semantic link discovery over relational data. In *Proceedings of the 18th international conference on World wide web*, pages 1101–1102. ACM, 2009.
11. Ian Horrocks. Ontologies and the semantic web. *CACM*, 51(11):58–67, 2008.
12. Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *International Conference of Computer Vision (ICCV)*, pages 1482–1489, 2005.
13. H. Newcombe, J. Kennedy, S. Axford, and A. James. Automatic linkage of vital records. *Science*, 130(3381):954–959, 1959.
14. Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
15. Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics*, 12:66–94, 2009.
16. Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1998.
17. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: a practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.
18. Heiko Stoermer and Nataliya Rassadko. Results of OKKAM feature based entity matching algorithm for instance matching contest of OAEI 2009. In *Proceedings of the ISWC 2009 workshop on ontology matching*, 2009.
19. Heiner Stuckenschmidt. A Semantic Similarity Measure for Ontology-Based Information. In *Proceedings of the 8th International Conference on Flexible Query Answering Systems*, 2009.
20. H.A. Taha. *Operations research: an introduction*. Prentice Hall New York, 2002.
21. G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, 2007.
22. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk–a link discovery framework for the web of data. In *2nd Linked Data on the Web Workshop*, 2009.
23. X. Zhang, Q. Zhong, F. Shi, J. Li, and J. Tang. RiMOM results for OAEI 2009. In *Proceedings of the ISWC 2009 workshop on ontology matching*, 2009.