Probabilistic-Logical Web Data Integration

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

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

Abstract. The integration of both distributed schemas and data repositories is a major challenge in data and knowledge management applications. Instances of this problem range from mapping database schemas to object reconciliation in the linked open data cloud. We present a novel approach to several important data integration problems that combines logical and probabilistic reasoning. We first provide a brief overview of some of the basic formalisms such as description logics and Markov logic that are used in the framework. We then describe the representation of the different integration problems in the probabilistic-logical framework and discuss efficient inference algorithms. For each of the applications, we conducted extensive experiments on standard data integration and matching benchmarks to evaluate the efficiency and performance of the approach. The positive results of the evaluation are quite promising and the flexibility of the framework makes it easily adaptable to other real-world data integration problems.

1 Introduction

The growing number of heterogeneous knowledge bases on the web has made data integration systems a key technology for sharing and accumulating distributed data and knowledge repositories. In this paper, we focus on (a) the problem of aligning description logic ontologies and (b) the problem of object reconciliation in open linked datasets[1].

Ontology matching, or ontology alignment, is the problem of determining correspondences between concepts, properties, and individuals of two or more different formal ontologies [12]. The alignment of ontologies allows semantic applications to exchange and enrich the data expressed in the respective ontologies. An important results of the yearly ontology alignment evaluation initiative (OAEI) [11,13] is that there is no single best approach to all existing matching problems. The factors influencing the quality of alignments range from differences in lexical similarity measures to variations in alignment extraction approaches. This insight provides justification not only for the OAEI itself but also for the

[1] The present chapter provides a more didactical exposition of the principles and methods presented in a series of papers of the same authors published in several conferences such as AAAI, UAI, and ESWC.

A. Polleres et al. (Eds.): Reasoning Web 2011, LNCS 6848, pp. 504–533, 2011.

development of a framework that facilitates the comparison of different strategies with a flexible and declarative formalism. We argue that Markov logic [39] provides and excellent framework for ontology matching. Markov logic (ML) offers several advantages over existing matching approaches. Its main strength is rooted in the ability to combine *soft* and *hard* first-order formulas. This allows the inclusion of both *known* logical and *uncertain* statements modeling potential correspondences and structural properties of the ontologies. For instance, hard formulas can help to reduce incoherence during the alignment process while soft formulas can factor in lexical similarity values computed for each correspondence. An additional advantage of ML is joint inference, that is, the inference of two or more interdependent hidden predicates. Several results show that joint inference is superior in accuracy when applied to a wide range of problems such as ontology refinement [53] and multilingual semantic role labeling [32].

Identifying different representations of the same data item is called object reconciliation. The problem of object reconciliation has been a topic of research for more than 50 years. It is also known as record linkage [14], entity resolution [3], and instance matching [15]. 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 such as the resource description framework (RDF). Using the proposed Markov logic based framework for data integration, we leverage schema information to exclude logically inconsistent correspondences between objects improving 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 approach to object reconciliation that is more accurate than state-of-the-art alignment systems.

We demonstrate how description logic axioms are modeled within the framework and show that alignment problems can be posed as linear optimization problems. These problems can be efficiently solved with integer linear programming methods also leveraging recent meta-algorithms such as cutting plane inference and delayed column generation first proposed in the context of Markov logic.

The chapter is organized as follows. First, we briefly introduce some basic formalism such as description logics and Markov logic. Second, we define ontology matching and object reconciliation and introduce detailed running examples that we use throughout the chapter to facilitate a deeper understanding of the ideas and methods. We also introduce the syntax and semantics of the ML framework and show that it can represent numerous different matching scenarios. We describe probabilistic reasoning in the framework of Markov logic and show that a solution to a given matching problem can be obtained by solving the maximum a-posteriori (MAP) problem of a ground Markov logic network using integer linear programming. We then report the results of an empirical evaluation of our method using some of the OAEI benchmark datasets.

2 Data Integration on the Web

The integration of distributed information sources is a key challenge in data and knowledge management applications. Instances of this problem range from mapping schemas of heterogeneous databases to object reconciliation in linked open data repositories. In the following, we discuss two instances of the data integration problem: ontology matching and object reconciliation. Both problems have been in the focus of the semantic web community in recent years. We investigate and assess the applicability and performance of our probabilistic-logical approach to data integration using these two prominent problems. In order to make the article comprehensive, however, we first briefly cover description logics and ontologies as these logical concepts are needed in later parts of the document.

2.1 Ontologies and Description Logics

An Ontology usually groups objects of the world that have certain properties in common (e.g. cities or countries) into concepts. A specification of the shared properties that characterize a set of objects is called a concept definition. Concepts can be arranged into a subclass–superclass relation in order to further discriminate objects into subgroups (e.g. capitals or European countries). Concepts can be defined in two ways, by enumeration of its members or by a concept expression. The specific logical operators that can be used to formulate concept expressions can vary between ontology languages.

Description logics are decidable fragments of first order logic that are designed to describe concepts in terms of complex logical expressions[2] The basic modeling elements in description logics are concepts (classes of objects), roles (binary relations between objects) and individuals (named objects). Based on these modeling elements, description logics contain operators for specifying so-called concept expressions that can be used to specify necessary and sufficient conditions for membership in the concept they describe. These modeling elements are provided with a formal semantics in terms of an abstract domain interpretation mapping $\mathcal{I}$ mapping each instance onto an element of an abstract domain $\Delta^{\mathcal{I}}$. Instances can be connected by binary relations defined as subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Concepts are interpreted as a subset of the abstract domain Δ. Intuitively, a concept is a set of instances that share certain properties. These properties are defined in terms of concept expressions. Typical operators are the Boolean operators as well as universal and existential quantification over relations to instances in other concepts.

A description logic knowledge base consists of two parts. The A-box contains information about objects, their type and relations between them, the so-called T-Box consists of a set of axioms about concepts (potentially defined in terms of complex concept expressions and relations. The first type of axioms can be used to describe instances. In particular, axioms can be used to state that an instance

[2] Details about the relation between description logics and first-order logic can be found in [4] and [51].

Table 1. Axiom patterns for representing description logic ontologies

DL Axiom	Semantics	Intuition
	A-Box	
$C(x)$	$x^{\mathcal{I}} \in C^{\mathcal{I}}$	x is of type C
$r(x,y)$	$(x^{\mathcal{I}}, y^{\mathcal{I}}) \in r^{\mathcal{I}}$	x is related to y by r
	T-Box	
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	C is more specific than D
$C \sqcap D \sqsubseteq \bot$	$C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$	C and D are disjoint
$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$	r is more specific than s
$r \equiv s^{-}$	$r^{\mathcal{I}} = \{(x,y) \mid (y,x) \in s^{\mathcal{I}}\}$	r is the inverse of s
$\exists r.\top \sqsubseteq C$	$(x^{\mathcal{I}}, y^{\mathcal{I}}) \in r^{\mathcal{I}} \Rightarrow x^{\mathcal{I}} \in C^{\mathcal{I}}$	the domain of r is restricted to C
$\exists r^{-1}.\top \sqsubseteq C$	$(x^{\mathcal{I}}, y^{\mathcal{I}}) \in r^{\mathcal{I}} \Rightarrow y^{\mathcal{I}} \in C^{\mathcal{I}}$	the range of r is restricted to C

belongs to a concept or that two instances are in a certain relation. It is easy to see, that these axioms can be used to capture case descriptions as labeled graphs. The other types of axioms describe relations between concepts and instances. It can be stated that one concept is a subconcept of the other (all its instances are also instances of this other concept). Further, we can define a relation to be a subrelation or the inverse of another relation. The formal semantics of concepts and relations as defined by the interpretation into the abstract domain $\Delta^{\mathcal{I}}$ can be used to automatically infer new axioms from existing definitions. Table 1 lists a few examples of DL axioms, their semantics, and the intuition behind them.

Encoding ontologies in description logics is beneficial, because it enables inference engines to reason about ontological definitions. In this context, deciding subsumption between two concept expressions, i.e. deciding whether one expression is more general than the other one is one of the most important reasoning tasks as it has been used to support various tasks including information integration [47], product and service matching [27] and query answering over ontologies [2].

2.2 Ontology Matching

Ontology matching is the process of detecting links between entities in heterogeneous ontologies. Based on a definition by Euzenat and Shvaiko [12], we formally introduce the notion of *correspondence* and *alignment* to refer to these links.

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 $q(\mathcal{O}_2)$. A correspondence between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a triple $\langle 3, e_1, e_2 \rangle\, r$ such that $e_1 \in q(\mathcal{O}_1)$, $e_2 \in q(\mathcal{O}_2)$, and r is a semantic relation. An alignment 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 and semantic relation. In the

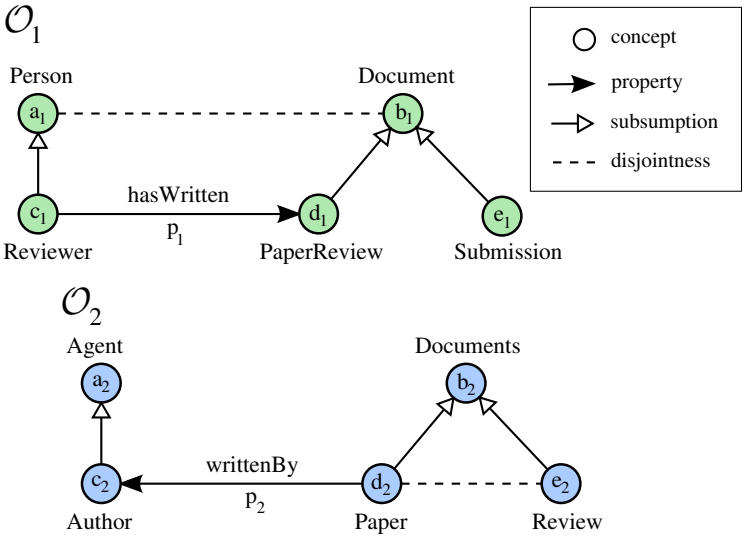


Fig. 1. Example ontology fragments

context of ontology matching, we are only interested in equivalence correspondences between concepts and properties. In the first step of the alignment process most matching systems compute a-priori similarities between matching candidates. These values are typically refined in later phases of the matching process. The underlying assumption is that the degree of similarity is indicative of the likelihood that two entities are equivalent. Given two matchable entities e_1 and e_2 we write $\sigma(e_1, e_2)$ to refer to this kind of a-priori similarity. Before presenting the formal matching framework, we motivate the approach by a simple instance of an ontology matching problem which we use as a running example.

Example 1. Figure 1 depicts fragments of two ontologies describing the domain of scientific conferences. The following axioms are part of ontology $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. If we apply a similarity measure σ based on the Levenshtein distance [26] there are four pairs of entities such that $\sigma(e_1, e_2) > 0.5$.

$$\sigma(Document, Documents) = 0.88 \quad (1)$$
$$\sigma(Reviewer, Review) = 0.75 \quad (2)$$
$$\sigma(hasWritten, writtenBy) = 0.7 \quad (3)$$
$$\sigma(PaperReview, Review) = 0.54 \quad (4)$$

The alignment consisting of these four correspondences contains two correct (1 & 4) and two incorrect (2 & 3) correspondences resulting in a precision of 50%.

Table 2. Discription logics axioms in the ontology of Figure 1

Ontology $\mathcal{O}_1$		Ontology $\mathcal{O}_2$	
$\exists hasWritten$	$\sqsubseteq Reviewer$	$\exists writtenBy$	$\sqsubseteq Paper$
$PaperReview$	$\sqsubseteq Document$	$Review$	$\sqsubseteq Documents$
$Reviewer$	$\sqsubseteq Person$	$Paper$	$\sqsubseteq Documents$
$Submission$	$\sqsubseteq Document$	$Author$	$\sqsubseteq Agent$
$Document$	$\sqsubseteq \neg Person$	$Paper$	$\sqsubseteq \neg Review$

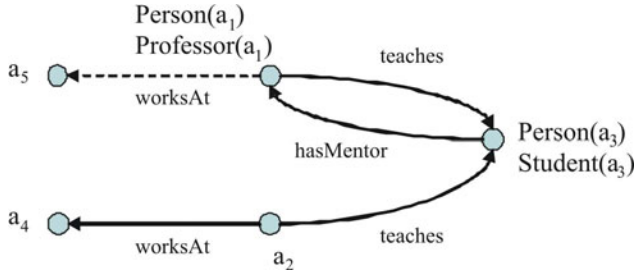
2.3 Object Reconciliation

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 [14], entity resolution [3], and instance matching [15]. 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 [12]. 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*.

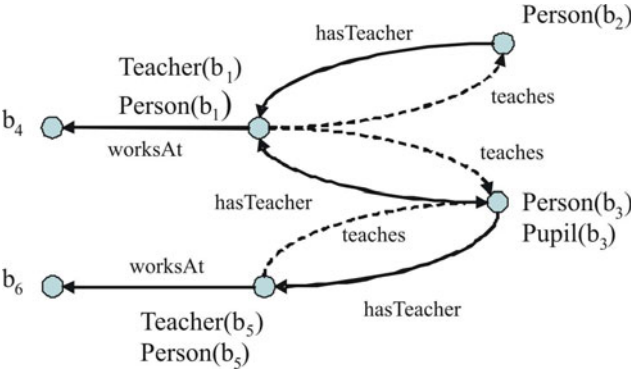
In the following we refer to an alignment that contains correspondences between concepts and properties as *terminological alignment* and to an alignment that contains correspondences between individuals as *instance alignment*. Since instance matching is the task of detecting pairs of instances that refer to the same real world object [15], 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 multiple sources of evidence.

Example 2. An A-box is a set of membership statements of the following form: $C(a), P(a, b)$ where a,b are constants, C is a concept name and P is a property name. Further, we extend the notion of an A-Box by also allowing membership statements of the form $\neg C(a)$ and $\neg P(a, b)$ stating that object a is not a member of Concept C and that the objects a and b are not in relation R, respectively. We illustrate the problem of object reconciliation using the following example A-Boxes and their corresponding graphs.

A-Boxes can be regarded as labeled directed multi-graphs, where object constants are represented by nodes and binary relations between objects are represented by links labeled with the name of the corresponding relation. Object reconciliation is the task of finding the 'right' mapping between the nodes in different A-Box graphs. The basis for finding the right mapping between different



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



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

Fig. 2. Examples of A-Boxes

objects is typically based on a measure of similarity between the nodes that is determined on the local or global structures in the corresponding graph. Typical features for determining the similarity of two objects are:

- the similarity of their labels
- the similarity of the classes the objects belong to
- the similarity of relations and related objects

Based on these features, we would generate a priori similarities. For the example we would receive high values for $\sigma(a_5, b_4)$, $\sigma(a_1, b_1)$, $\sigma(a_3, b_3)$, $\sigma(a_3, b_2)$, $\sigma(a_2, b_5)$ and $\sigma(a_4, b_6)$. Besides the similarity between objects, in the case where the A-Box is based on an ontology, the logical constraints from the ontologies should be taken into account in the matching process. In particular, objects should not be maps on each other if they have incompatible types. In the example this means that assuming the underlying ontology contains a statement $student \perp pupil$ declaring the classes 'student' and 'pupil' as disjoint, the objects a_3 and b_3 should not be mapped on each other, despite the high a priori similarity.

3 Probabilistic-Logical Languages and Ontologies

Data integration for heterogeneous knowledge bases typically involves both purely logical and uncertain data. For instance, the description logic axioms of the ontologies are known to be true and, therefore, should be modeled as logical rules – the alignment system should not alter the logical structure of the input ontologies. Conversely, matching systems usually rely on degrees of confidence that have been derived through the application of lexical similarity, data mining, and machine learning algorithms. The presence of both known logical rules and degrees of uncertainty requires formalism that allow the representation of both deterministic and uncertain aspects of the problem. In the following, we introduce such a probabilistic-logical framework based on Markov logic and show how description logic ontologies are represented in the language. Moreover, we describe the application of an efficient probabilistic inference algorithm that uses integer linear programming.

3.1 Markov Logic

Markov logic combines first-order logic and undirected probabilistic graphical models [39]. A Markov logic network (MLN) is a set of first-order formulas with weights. Intuitively, the more evidence we have that a formula is true the higher the weight of this formula. To simplify the presentation of the technical parts we do *not* include functions. In addition, we assume that all (ground) formulas of a Markov logic network are in clausal form and use the terms *formula* and *clause* interchangeably.

Syntax. A signature is a triple $S = (O, H, C)$ with O a finite set of observable predicate symbols, H a finite set of hidden predicate symbols, and C a finite set of constants. A Markov logic network (MLN) is a set of pairs $\{(F_i, w_i)\}$ with each F_i being a function-free first-order formula built using predicates from $O \cup H$ and each $w_i \in \mathbb{R}$ a real-valued weight associated with formula F_i. We can represent hard constraints using large weights.

Semantics. Let $M = (F_i, w_i)$ be a Markov logic network with signature $S = (O, H, C)$. A *grounding* of a first-order formula F is generated by substituting each occurrence of every variable in F with constants in C. Existentially quantified formulas are substituted by the disjunctions of their groundings over the finite set of constants. A formula that does not contain any variables is *ground*. A formula that consists of a single predicate is an *atom*. Note that Markov logic makes several assumptions such as (a) different constants refer to different objects and (b) the only objects in the domain are those representable using the constants [39]. A set of ground atoms is a *possible world*. We say that a possible world W *satisfies* a formula F, and write $W \models F$, if F is true in W. Let $\mathcal{G}_F^C$ be the set of all possible groundings of formula F with respect to C. We say that W satisfies $\mathcal{G}_F^C$, and write $W \models \mathcal{G}_F^C$, if F satisfies every formula in $\mathcal{G}_F^C$. Let $\mathcal{W}$

be the set of all possible worlds with respect to S. Then, the probability of a possible world W is given by

$$p(W) = \frac{1}{Z} \exp\left(\sum_{(F_i, w_i)} \sum_{G \in \mathcal{G}_{F_i}^C :\ W \models G} w_i \right).$$

Here, Z is a normalization constant. The score s_W of a possible world W is the sum of the weights of the ground formulas implied by W

$$s_W = \sum_{(F_i, w_i)} \sum_{G \in \mathcal{G}_{F_i}^C :\ W \models G} w_i. \tag{5}$$

We will see later that, in the data integration context, possible worlds correspond to possible alignments. Hence, the problem of deriving the most probably alignment given the evidence can be interpreted as finding the possible world W with highest score.

3.2 Representing Ontologies and Alignments in Markov Logic

Our approach for data integration based on logics and probability is now based on the idea of representing description logic ontologies as Markov logic networks and utilizing the weights to incorporate similarity scores into the integration process [34]. The most obvious way to represent a description logic ontology in Markov logic would be to directly use the first-order translation of the ontology. For instance, the axiom $C \sqsubseteq D$ would be written as $\forall x\ C(x) \Rightarrow D(x)$. In other words, the representation would simply map between concepts and unary predicates and roles and binary predicates. However, we take a *different* approach by mapping axioms to predicates and use constants to represent the classes and relations in the ontology. Some typical axioms with their respective predicates are the following:

$$\begin{array}{ll} C \sqsubseteq D & \mapsto sub(c, d) \\ C \sqcap D \sqsubseteq \bot & \mapsto dis(c, d) \\ \exists r.T \sqsubseteq C & \mapsto dom(r, c) \\ \exists r^{-1}.T \sqsubseteq C & \mapsto range(r, c) \end{array}$$

This way of representing description logic ontologies has the advantage that we can model some basic inference rules and directly use them in the probabilistic reasoning process. For example, we can model the transitivity of the subsumption relation as

$$sub(x, y) \wedge sub(y, z) \Rightarrow sub(x, z)$$

and the fact that two classes that subsume each other cannot be disjoint at the same time

$$\neg sub(x, y) \vee \neg dis(x, y)$$

While the use of such axioms in a Markov logic network does not guarantee consistency and coherence of the results, they often cover the vast majority of

Table 3. The description logic $\mathcal{EL}^{++}$ without nominals and concrete domains

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ ... \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ ... \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

conflicts that can exist in an ontology, especially in cases where the ontology is rather simple and does not contain a complex axiomatization.

For certain description logics, it is possible to completely capture the model using the kind of translation described above. In particular, if an ontology can be reduced to a normal form with a limited number of axiom types, we can provide a complete translation based on this normal form. An example for such a description logic is $\mathcal{EL}^{++}$, a light weight description logic that supports polynomial time reasoning. Table 3 shows the types of axioms an $\mathcal{EL}^{++}$ Model can be reduced to.

We can completely translation any $\mathcal{EL}^{++}$ model into a Markov Logic representation using the following translation rules:

$$\begin{aligned}
C_1 \sqsubseteq D &\mapsto sub(c_1, d)\\
C_1 \sqcap C_2 \sqsubseteq D &\mapsto int(c_1, c_2, d)\\
C_1 \sqsubseteq \exists r.C_2 &\mapsto rsup(c_1, r, c_2)\\
\exists r.C_1 \sqsubseteq D &\mapsto rsub(c_1, r, d)\\
r \sqsubseteq s &\mapsto psub(r, s)\\
r_1 \circ r_2 \sqsubseteq r_3 &\mapsto pcom(r_1, r_2, r_3)
\end{aligned}$$

In principle, such a complete translation is possible whenever there is a normal form representation of a description logic that reduces the original model to a finite number of axiom types that can be captured by a respective predicate in the Markov logic network.

Finally, being interested in data integration, we often treat correspondences between elements from different models separately although in principle they could be represented by ordinary DL axioms. In particular, we often use the following translation of correspondences to weighted ground predicates of the Markov logic network

$$(e_1, e_2, R, c) \mapsto \langle map_R(e_1, e_2), c \rangle$$

where c is a a-priori confidence values.

3.3 MAP Inference and Integer Linear Programming

If we want to determine the most probable state of a MLN, we need to compute the set of ground atoms of the hidden predicates that maximizes the probability given both the ground atoms of observable predicates and all ground formulas. This is an instance of MAP (maximum a-posteriori) inference in the ground Markov logic network. Let $\mathbf{O}$ be the set of all ground atoms of observable predicates and $\mathbf{H}$ be the set of all ground atoms of hidden predicates both with respect to C. We make the closed world assumption with respect to the observable predicates. Assume that we are given a set $\mathbf{O'} \subseteq \mathbf{O}$ of ground atoms of observable predicates. In order to find the most probable state of the MLN we have to compute

$$\underset{\mathbf{H'} \subseteq \mathbf{H}}{\operatorname{argmax}} \sum_{(F_i, w_i)} \sum_{G \in \mathcal{G}_{F_i}^C \colon \mathbf{O'} \cup \mathbf{H'} \models G} w_i.$$

Every $\mathbf{H'} \subseteq \mathbf{H}$ is called a *state*. It is the set of *active* ground atoms of hidden predicates. Markov logic is by definition a declarative language, separating the formulation of a problem instance from the algorithm used for probabilistic inference. MAP inference in Markov logic networks is essentially equivalent to the weighted MAX-SAT problem and, therefore, NP-hard. Integer linear programming (ILP) is an effective method for solving exact MAP inference in undirected graphical models [41,50] and specifically in Markov logic networks [40]. ILP is concerned with optimizing a linear objective function over a finite number of integer variables, subject to a set of linear constraints over these variables [43]. We omit the formal details of the ILP representation of a MAP problem and refer the reader to [40].

Example 3. Consider a small instance of the ontology alignment problem which involves both soft and hard formulas. ML was successfully applied to ontology matching problems in earlier work [34]. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be the two ontologies in Figure 3 with the (a-priori computed) string similarities between the concept labels given in Table 4. Let $S = (O, H, C)$ be the signature of a MLN M with $O = \{sub_1, sub_2, dis_1, dis_2\}$, $H = \{map\}$, and $C = \{a_1, b_1, c_1, a_2, b_2\}$. Here, the observable predicates model the subsumption and disjointness relationships between concepts C in the two ontologies and map is the hidden predicate modeling the sought-after matching correspondences. We also assume that the predicates are typed meaning that, for instance, valid groundings of $map(x, y)$ are those with $x \in \{a_1, b_1, c_1\}$ and $y \in \{a_2, b_2\}$. Furthermore, let us assume that the MLN M includes the following formula with weight $w = 10.0$:

$$\forall x, x', y, y' : dis_1(x, x') \wedge sub_2(y, y') \Rightarrow (\neg map(x, y) \vee \neg map(x', y'))$$

The formula makes those alignments less likely that match concepts x with y and x' with y' if x is disjoint with x' in the first ontology and y' subsumes y in the second. We also include cardinality formulas with weight 10.0 forcing alignments to be one-to-one and functional:

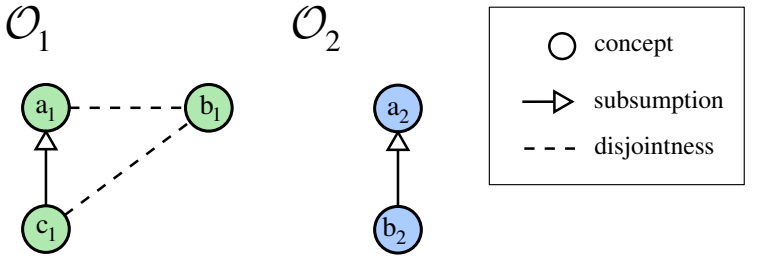


Fig. 3. Small fragments of two ontologies

Table 4. A-priori similarities between concept labels

	a_1	b_1	c_1
a_2	0.95	0.25	0.12
b_2	0.55	0.91	0.64

$$\forall x, y, z : map(x, y) \wedge map(x, z) \Rightarrow y = z$$
$$\forall x, y, z : map(x, y) \wedge map(z, y) \Rightarrow x = z$$

In addition, we add the formulas $map(x, y)$ with weight $\sigma(x, y)$ for all $x \in \{a_1, b_1, c_1, d_1\}$ and $y \in \{a_2, b_2\}$ where $\sigma(x, y)$ is the label similarity from Table 4. The observed ground atoms are $sub_1(c_1, a_1), dis_1(a_1, b_1), dis_1(b_1, a_1), dis_1(b_1, c_1), dis_1(c_1, b_1)$ for ontology $\mathcal{O}_1$ and $sub_2(b_2, a_2)$ for ontology $\mathcal{O}_2$. This results in the following relevant ground formulas for the coherence reducing constraint where each observable predicates has been substituted with its observed value:

$$\neg map(a_1, b_2) \vee \neg map(b_1, a_2) \quad (6)$$
$$\neg map(b_1, b_2) \vee \neg map(a_1, a_2) \quad (7)$$
$$\neg map(b_1, b_2) \vee \neg map(c_1, a_2) \quad (8)$$
$$\neg map(c_1, b_2) \vee \neg map(b_1, a_2) \quad (9)$$

For instance, the ground formulas (2) is encoded in an ILP by introducing a new binary variable y which is added to the objective function with coefficient 10.0 and, in addition, by introducing the following linear constraints enforcing the value of y to be equivalent to the truth value of the formula:

$$-x_{a,b} - y \leq -1$$
$$-x_{b,a} - y \leq -1$$
$$x_{a,b} + x_{b,a} + y \leq 2$$

The binary ILP variables $x_{a,b}$ and $x_{b,a}$ correspond to ground atoms $map(a_1, b_2)$ and $map(b_1, a_2)$, respectively. The ILP for our small example includes 19 variables (columns) and 39 linear constraints (12 from the coherence and 27 from the

cardinality formulas) which we omit due to space considerations. The preprocessing step of grounding only those clauses that can evaluate to *false* given the current state of observable variables is similar to the approach presented in [44]. The ILP optimizations used for the inference procedures are not the focus of this article and we refer the reader to [40] and [33] for the details. However, in the following section we will show a typical matching formalization in Markov logic, the resulting ground formulas, and the corresponding integer linear program.

4 Markov Logic and Ontology Matching

We provide a formalization of the ontology matching problem within the probabilistic-logical framework. The presented approach has several advantages over existing methods such as ease of experimentation, incoherence mitigation during the alignment process, and the incorporation of a-priori confidence values. We show empirically that the approach is efficient and more accurate than existing matchers on an established ontology alignment benchmark dataset.

4.1 Problem Representation

Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and an initial a-priori similarity σ we apply the following formalization. First, we introduce observable predicates O to model the structure of $\mathcal{O}_1$ and $\mathcal{O}_2$ with respect to both concepts and properties. For the sake of simplicity we use uppercase letters D, E, R to refer to individual concepts and properties in the ontologies and lowercase letters d, e, r to refer to the corresponding constants in C. In particular, we add ground atoms of observable predicates to $\mathcal{F}^h$ for $i \in \{1, 2\}$ according to the following rules:

$$
\begin{aligned}
\mathcal{O}_i &\models D \sqsubseteq E && \mapsto && sub_i(d, e) \\
\mathcal{O}_i &\models D \sqsubseteq \neg E && \mapsto && dis_i(d, e) \\
\mathcal{O}_i &\models \exists R.\top \sqsubseteq D && \mapsto && sub_i^d(r, d) \\
\mathcal{O}_i &\models \exists R^{-1}.\top \sqsubseteq D && \mapsto && sub_i^r(r, d) \\
\mathcal{O}_i &\models \exists R.\top \sqsupseteq D && \mapsto && sup_i^d(r, d) \\
\mathcal{O}_i &\models \exists R^{-1}.\top \sqsupseteq D && \mapsto && sup_i^r(r, d) \\
\mathcal{O}_i &\models \exists R.\top \sqsubseteq \neg D && \mapsto && dis_i^d(r, d) \\
\mathcal{O}_i &\models \exists R^{-1}.\top \sqsubseteq \neg D && \mapsto && dis_i^r(r, d)
\end{aligned}
$$

The knowledge encoded in the ontologies is assumed to be true. Hence, the ground atoms of observable predicates are added to the set of hard constraints $\mathcal{F}^h$, making them hold in every computed alignment. The hidden predicates map_c and map_p, on the other hand, model the sought-after concept and property correspondences, respectively. Given the state of the observable predicates, we are interested in determining the state of the hidden predicates that maximize the a-posteriori probability of the corresponding possible world. The ground

atoms of these hidden predicates are assigned the weights specified by the a-priori similarity σ. The higher this value for a correspondence the more likely the correspondence is correct *a-priori*. Hence, the following ground formulas are added to $\mathcal{F}^s$, the set of soft formulas:

$$\begin{array}{ll} (map_c(c,d), & \sigma(C,D)) \quad \text{if C and D are concepts} \\ (map_p(p,r), & \sigma(P,R)) \quad \text{if P and R are properties} \end{array}$$

Notice that the distinction between m_c and m_p is required since we use typed predicates and distinguish between the *concept* and *property* type.

Cardinality Constraints. A method often applied in real-world scenarios is the selection of a functional one-to-one alignment [7]. Within the ML framework, we can include a set of hard cardinality constraints, restricting the alignment to be functional and one-to-one. In the following we write x, y, z to refer to variables ranging over the appropriately typed constants and omit the universal quantifiers.

$$map_c(x,y) \wedge map_c(x,z) \Rightarrow y = z$$
$$map_c(x,y) \wedge map_c(z,y) \Rightarrow x = z$$

Analogously, the same formulas can be included with hidden predicates map_p, restricting the property alignment to be one-to-one and functional.

Coherence Constraints. Incoherence occurs when axioms in ontologies lead to logical contradictions. Clearly, it is desirable to avoid incoherence during the alignment process. Some methods of incoherence removal for ontology alignments were introduced in [30]. All existing approaches, however, remove correspondences *after* the computation of the alignment. Within the ML framework we can incorporate incoherence reducing constraints *during* the alignment process for the first time. This is accomplished by adding formulas of the following type to $\mathcal{F}^h$, the set of hard formulas.

$$dis_1(x,x') \wedge sub_2(x,x') \Rightarrow \neg(map_c(x,y) \wedge map_c(x',y'))$$
$$dis_1^d(x,x') \wedge sub_2^d(y,y') \Rightarrow \neg(map_p(x,y) \wedge map_c(x',y'))$$

The second formula, for example, has the following purpose. Given properties X, Y and concepts X', Y'. Suppose that $\mathcal{O}_1 \models \exists X.\top \sqsubseteq \neg X'$ and $\mathcal{O}_2 \models \exists Y.\top \sqsubseteq Y'$. Now, if $\langle X, Y, \equiv\rangle$ and $\langle X', Y', \equiv\rangle$ were both part of an alignment the merged ontology would entail both $\exists X.\top \sqsubseteq X'$ and $\exists X.\top \sqsubseteq \neg X'$ and, therefore, $\exists X.\top \sqsubseteq \bot$. The specified formula prevents this type of incoherence. It is known that such constraints, if carefully chosen, can avoid a majority of possible incoherences [29].

Stability Constraints. Several existing approaches to schema and ontology matching propagate alignment evidence derived from structural relationships between concepts and properties. These methods leverage the fact that existing

evidence for the equivalence of concepts C and D also makes it more likely that, for example, child concepts of C and child concepts of D are equivalent. One such approach to evidence propagation is *similarity flooding* [31]. As a reciprocal idea, the general notion of stability was introduced, expressing that an alignment should not introduce new structural knowledge [28]. The *soft* formula below, for instance, decreases the probability of alignments that map concepts X to Y and X' to Y' if X' subsumes X but Y' does *not* subsume Y.

$$\langle sub_1(x, x') \wedge \neg sub_2(y, y') \Rightarrow map_c(x, y) \wedge map_c(x', y'),\ w_1 \rangle$$
$$\langle sub_1^d(x, x') \wedge \neg sub_2^d(y, y') \Rightarrow map_p(x, y) \wedge map_c(x', y'),\ w_2 \rangle$$

Here, w_1 and w_2 are *negative* real-valued weights, rendering alignments that satisfy the formulas possible but less likely.

The presented list of cardinality, coherence, and stability constraints is by no means meant to be exhaustive. Other constraints could, for example, model known correct correspondences or generalize the one-to-one alignment to m-to-n alignments. Moreover, a novel hidden predicate could be added modeling correspondences between instances of the ontologies. To keep the discussion of the approach simple, however, we leave these considerations to future research.

Example 4. We apply the previous formalization to Example 1. To keep it simple, we only use a-priori values, cardinality, and coherence constraints. Given the two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ in Figure 1, and the matching hypotheses (1) to (4) from Example 1, the ground MLN would include the following relevant ground formulas. We use the concept and property labels from Figure 1 and omit ground atoms of observable predicates.

A-priori similarity

$$\langle map_c(b_1, b_2), 0.88 \rangle, \langle map_c(c_1, e_2), 0.75 \rangle, \langle map_p(p_1, p_2), 0.7 \rangle, \langle map_c(d_1, e_2), 0.54 \rangle$$

Cardinality constraints

$$map_c(c_1, e_2) \wedge map_c(d_1, e_2) \Rightarrow c_1 = d_1 \tag{10}$$

Coherence constraints

$$dis_1^d(p_1, b_1) \wedge sub_2^d(p_2, b_2) \Rightarrow \neg(map_p(p_1, p_2) \wedge map_c(b_1, b_2)) \tag{11}$$
$$dis_1(b_1, c_1) \wedge sub_2(b_2, e_2) \Rightarrow \neg(map_c(b_1, b_2) \wedge map_c(c_1, e_2)) \tag{12}$$
$$sub_1^d(p_1, c_1) \wedge dis_2^d(p_2, e_2) \Rightarrow \neg(map_p(p_1, p_2) \wedge map_c(c_1, e_2)) \tag{13}$$

Let the binary ILP variables x_1, x_2, x_3, and x_4 model the ground atoms $map_c(b_1, b_2), map_c(c_1, e_2), map_p(p_1, p_2)$, and $map_c(d_1, e_2)$, respectively. The set of ground formulas is then encoded in the following integer linear program:

Maximize: $0.88x_1 + 0.75x_2 + 0.7x_3 + 0.54x_4$

Subject to

$$x_2 + x_4 \leq 1 \tag{14}$$

$$x_1 + x_3 \leq 1 \tag{15}$$

$$x_1 + x_2 \leq 1 \tag{16}$$

$$x_2 + x_3 \leq 1 \tag{17}$$

The a-priori confidence values of the potential correspondences are factored in as coefficients of the objective function. Here, the ILP constraint (9) corresponds to ground formula (5), and ILP constraints (10),(11), and (12) correspond to the coherence ground formulas (6), (7), and (8), respectively. An optimal solution to the ILP consists of the variables x_1 and x_4 corresponding to the correct alignment $\{m_c(b_1, b_2), m_c(d_1, e_2)\}$. Compare this with the alignment $\{map_c(b_1, b_2), map_c(c_1, e_2), map_p(p_1, p_2)\}$ which would be the outcome without coherence constraints.

4.2 Experiments

We use the Ontofarm dataset [49] as basis for our experiments. It is the evaluation dataset for the OAEI conference track which consists of several ontologies modeling the domain of scientific conferences [11]. The ontologies were designed by different groups and, therefore, reflect different conceptualizations of the same domain. Reference alignments for seven of these ontologies are made available by the organizers. These 21 alignments contain correspondences between concepts and properties including a reasonable number of non-trivial cases. For the a-priori similarity σ we decided to use a standard lexical similarity measure. After converting the concept and object property names to lowercase and removing delimiters and stop-words, we applied a string similarity measure based on the Levensthein distance. More sophisticated a-priori similarity measures could be used but since we want to evaluate the benefits of the ML framework we strive to avoid any bias related to custom-tailored similarity measures. We applied the reasoner Pellet [45] to create the ground MLN formulation and used The-Beast[3] [40] to convert the MLN formulations to the corresponding ILP instances. Finally, we applied the mixed integer programming solver SCIP[4] to solve the ILP. All experiments were conducted on a desktop PC with AMD Athlon Dual Core Processor 5400B with 2.6GHz and 1GB RAM. The software as well as additional experimental results are available at `http://code.google.com/p/ml-match/`.

The application of a threshold τ is a standard technique in ontology matching. Correspondences that match entities with high similarity are accepted while correspondences with a similarity less than τ are deemed incorrect. We evaluated our approach with thresholds on the a-priori similarity measure σ ranging from 0.45 to 0.95. After applying the threshold τ we normalized the values to the range $[0.1, 1.0]$. For each pair of ontologies we computed the F_1-value, which is the harmonic mean of precision and recall, and computed the mean of this value over all 21 pairs of ontologies. We evaluated four different settings:

[3] `http://code.google.com/p/thebeast/`

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

- **ca**: The formulation includes only cardinality constraints.
- **ca+co**: The formulation includes only cardinality and coherence constraints.
- **ca+co+sm**: The formulation includes cardinality, coherence, and stability constraint, and the weights of the stability constraints are determined manually. Being able to set *qualitative* weights manually is crucial as training data is often unavailable. The employed stability constraints consist of (1) constraints that aim to guarantee the stability of the concept hierarchy, and (2) constraints that deal with the relation between concepts and property domain/range restrictions. We set the weights for the first group to -0.5 and the weights for the second group to -0.25. This is based on the consideration that subsumption axioms between concepts are specified by ontology engineers more often than domain and range restriction of properties [10]. Thus, a pair of two correct correspondences will less often violate constraints of the first type than constraints of the second type.
- **ca+co+sl**: The formulation also includes cardinality, coherence, and stability constraint, but the weights of the stability constraints are learned with a simple online learner using the perceptron rule. During learning we fixed the a-priori weights and learned only the weights for the stability formulas. We took 5 of the 7 ontologies and learned the weights on the 10 resulting pairs. With these weights we computed the alignment and its F_1-value for the remaining pair of ontologies. This was repeated for each of the 21 possible combinations to determine the mean of the F_1-values.

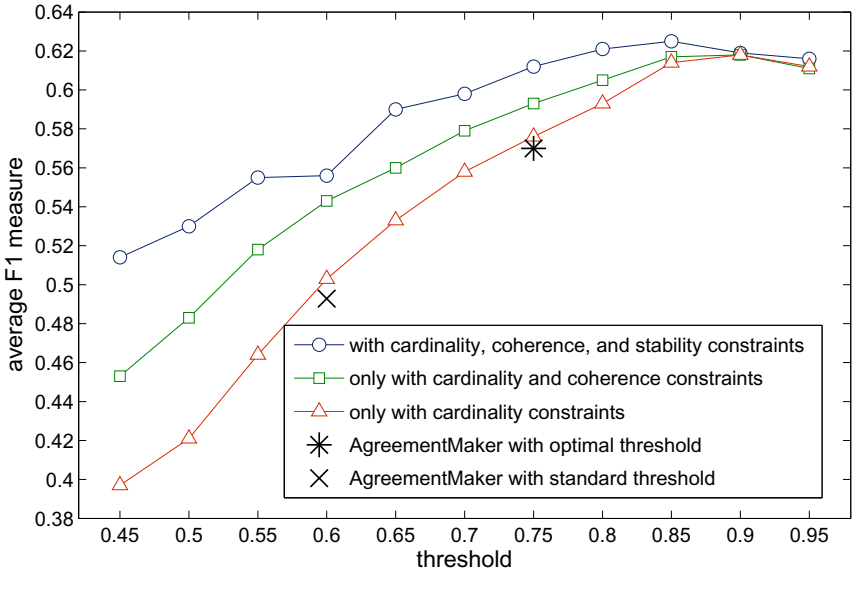


Fig. 4. F_1-values for **ca**, **ca+co**, and **ca+co+sm** averaged over the 21 OAEI reference alignments for thresholds ranging from 0.45 to 0.95. AgreementMaker was the best performing system on the conference dataset of the latest ontology evaluation initiative in 2009.

The lower the threshold the more complex the resulting ground MLN and the more time is needed to solve the corresponding ILP. The average time needed to compute one alignment was 61 seconds for $\tau = 0.45$ and 0.5 seconds for $\tau = 0.85$. Figure 4 depicts the average F_1-values for **ca**, **ca+co**, and **ca+co+sm** compared to the average F_1-values achieved by AgreementMaker [7], the best-performing system in the OAEI conference track of 2009. These average F_1-values of AgreementMaker were obtained using two different thresholds. The first is the default threshold of AgreementMaker and the second is the threshold at which the average F_1-value attains its maximum.

The inclusion of coherence constraints (**ca+co**) improves the average F_1-value of the alignments for low to moderate thresholds by up to 6% compared to the **ca** setting. With increasing thresholds this effect becomes weaker and is negligible for $\tau \geq 0.9$. This is the case because alignments generated with **ca** for thresholds ≥ 0.9 contain only a small number of incorrect correspondences. The addition of stability constraints (**ca+co+sm**) increases the quality of the alignments again by up to 6% for low to moderate thresholds. In the optimal configuration (**ca+co+sl** with $\tau = 0.85$) we measured an average F_1-value of 0.63 which is a 7% improvement compared to AgreementMaker's 0.56. What is more important to understand, however, is that our approach generates more accurate results over a wide range of thresholds and is therefore more robust to threshold estimation. This is advantageous since in most real-world matching scenarios the estimation of appropriate thresholds is not possible. While the **ca** setting generates F_1-values > 0.57 for $\tau \geq 0.75$ the **ca+co+sm** setting generates F_1-values > 0.59 for $\tau \geq 0.65$. Even for $\tau = 0.45$, usually considered an inappropriate threshold choice, we measured an average F_1-value of 0.51 and average precision and recall values of 0.48 and 0.60, respectively. Table 5 compares the average F_1-values of the ML formulation (a) with manually set weights for the stability constraints, (b) with learned weights for the stability constraints, and (c) without any stability constraints. The values indicate that using stability constraints improves alignment quality with both learned and manually set weights.

Table 5. Average F_1-values over the 21 OAEI reference alignments for manual weights (ca+co+sm) vs. learned weights (ca+co+sl) vs. formulation without stability constraints (ca+co); thresholds range from 0.6 to 0.95.

threshold	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
ca+co+sm	0.56	**0.59**	**0.60**	**0.61**	**0.62**	**0.63**	0.62	**0.62**
ca+co+sl	**0.57**	0.58	0.58	**0.61**	0.61	0.61	**0.63**	**0.62**
ca+co	0.54	0.56	0.58	0.59	0.61	0.62	0.62	0.61

5 Markov Logic and Object Reconciliation

We are primarily concerned with the scenario where both A-Boxes are described in terms of the same T-Box. The presented approach does not rely on specific

types of axioms or a set of predefined rules but on a well defined semantic similarity measure. In particular, our approach is based on the measure proposed by Stuckenschmidt [48]. 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 $\mathcal{T}$.

5.1 Problem Representation

The current instance matching configuration leverages terminological structure and combines it with lexical similarity measures. The approach is presented in more detail in [37]. The alignment system uses one T-Box $\mathcal{T}$ but two different A-Boxes $\mathcal{A}_1 \in \mathcal{O}_1$ and $\mathcal{A}_2 \in \mathcal{O}_2$. In cases with two different T-Boxes the T-Box matching approach is applied as a preprocessing step to merge the two aligned T-Boxes first. The approach offers complete conflict elimination meaning that the resulting alignment is always consistent for OWL DL ontologies. To enforce consistency, we need to add constraints to model conflicts, that is, we have to prevent an equivalence correspondence between two individuals if there exists a positive class assertion for the first individual and a negative for the second for the same class. These constraints are incorporated for both property and concept assertions. Analogous to the concept and property alignment before, we introduce the hidden predicate map_i representing instance correspondences. Let C be a concept and P be a property of T-Box $\mathcal{T}$. Further, let $A \in \mathcal{A}_1$ and $B \in \mathcal{A}_2$ be individuals in the respective A-Boxes. Then, using a reasoner such as Pellet, ground atoms are added to the set of *hard* constraints $\mathcal{F}^h$ according to the following rules:

$$
\begin{array}{lll}
\mathcal{T} \cup \mathcal{A}_1 \models C(A) & \wedge\ \mathcal{T} \cup \mathcal{A}_2 \models \neg C(B) & \mapsto\ \neg map_i(a,b) \\
\mathcal{T} \cup \mathcal{A}_1 \models \neg C(A) & \wedge\ \mathcal{T} \cup \mathcal{A}_2 \models C(B) & \mapsto\ \neg map_i(a,b) \\
\mathcal{T} \cup \mathcal{A}_1 \models P(A,A') & \wedge\ \mathcal{T} \cup \mathcal{A}_2 \models \neg P(B,B') & \mapsto\ \neg map_i(a,b) \vee \neg map_i(a',b') \\
\mathcal{T} \cup \mathcal{A}_1 \models \neg P(A,A') & \wedge\ \mathcal{T} \cup \mathcal{A}_2 \models P(B,B') & \mapsto\ \neg map_i(a,b) \vee \neg map_i(a',b')
\end{array}
$$

In addition to these formulas we included cardinality constraints analogous to those used in the previous concept and property alignment problem. In the instance matching formulation, the a-priori similarity $\sigma_{\mathbf{c}}$ and $\sigma_{\mathbf{p}}$ measures the *normalized overlap* of concept and property assertions, respectively. For more details on these measures, we refer the reader to [37]. The following formulas are added to the set of soft formulas $\mathcal{F}^s$:

$$
\begin{array}{rl}
\langle map_i(a,b),\ \ \sigma_{\mathbf{c}}(A,B)\rangle & \text{if A and B are instances} \\
\langle map_i(a,b) \wedge map_i(c,d),\ \ \sigma_{\mathbf{p}}(A,B,C,D)\rangle & \text{if A, B, C, and D are instances}
\end{array}
$$

Algorithm 1. $\sigma(entity_1, entity_2)$

```
if entity1 and entity2 are either concepts or properties then
  value ← 0
  for all Values s1 of URI, labels, and OBOtoOWL constructs in entity1 do
    for all Values s2 of URI, labels, and OBOtoOWL constructs in entity1 do
      value ← Max(value, sim(s1, s2))
    end for
  end for
  return value
end if
if entity1 and entity2 are individuals then
  Map⟨URI, double⟩ similarities ← null
  for all dataproperties dp1 of entity1 do
    uri1 ← URI of dp1
    for all dataproperties dp2 of entity2 do
      if uri1 equals URI of dp2 then
        value ← sim(valueofdp1, valueofdp2)
        if uri1 is entailed in similarities then
          update entry ⟨uri1, old_value⟩ to ⟨uri1, Minimum (old_value+value, 1)⟩
          in similarities
        else
          add new entry pair ⟨uri1, value⟩ in similarities
        end if
      end if
    end for
  end for
  return (sum of all values in similarities)/(length of similarities)
end if
```

5.2 Similarity Computation

Algorithm 1 was used for computing the a-priori similarity $\sigma(entity_1, entity_2)$. In the case of concept and property alignments, the a-priori similarity is computed by taking the maximal similarity between the URIs, labels and *OBO to OWL* constructs. In case of instance matching the algorithm goes through all data properties and takes the average of the similarity scores.

5.3 Experiments

The IIMB benchmark is a semi-automatically generated benchmark for instance matching. IIMB 2010 is created by extracting individuals from Freebase[5], an open knowledge base that contains information about 11 million real objects including movies, books, TV shows, celebrities, locations, companies and more. Data extraction has been performed using the query language JSON together with the Freebase JAVA API[6]. From this large dataset, 29 concepts, 20 object

[5] `http://www.freebase.com/`

[6] `http://code.google.com/p/freebase-java/`

properties, 12 data properties and a fraction of their underlying data have been chosen for the benchmark. The benchmark has been generated in a small version consisting of 363 individuals and in a large version containing 1416 individuals, respectively. Furthermore, the dataset consists of 80 different test cases divided into 4 sets of 20 test cases each. These sets have been designed according to the Semantic Web INstance Generation (SWING) approach presented in [16]. In the following, we will explain the SWING approach and its different transformation techniques resulting in the 80 different test cases in more detail.

Data acquisition techniques. SWING provides a set of techniques for the acquisition of data from the repositories of linked data and their representation as a reference OWL ABox. In SWING, we work on open repositories by addressing two main problems featuring this kind of data sources. First, we support the evaluation designer in defining a subset of data by choosing both the data categories of interest and the desired size of the benchmark. Second, in the data enrichment activity, we add semantics to the data acquired. In particular, we adopt specific ontology design patterns that drive the evaluation designer in defining a data description scheme capable of supporting the simulation of a wide spectrum of data heterogeneities. These techniques include

- adding super classes and super properties,
- converting attributes to class assertions,
- determining and adding new disjointness restrictions,
- enriching the ontology with additional inverse properties, and
- specifying additional domain and range restrictions.

Data transformation techniques. In the subsequent *data transformation* process the TBox is unchanged, while the ABox is modified in several ways by generating a set of new ABoxes, called *test cases*. Each test case is produced by transforming the individual descriptions in the reference ABox in new individual descriptions that are inserted in the test case at hand. The goal of transforming the original individuals is twofold: on one hand, we provide a simulated situation where data referred to the same objects are provided in different data sources; on the other hand, we generate a number of datasets with a variable degree of data quality and complexity.

The applied transformation techniques are categorized as followed:

- *Data value transformation* operations work on the concrete values of data properties and their datatypes when available. The output is a new concrete value. This category has been applied to the test cases 1-20 of the IIMB 2010 benchmark.
- *Data structure transformation* operations change the way data values are connected to individuals in the original ontology graph and change the type and number of properties associated with a given individual. They are implemented in the transformations 21-40 of the IIMB 2010 benchmark.
- *Data semantic transformation* operations are based on the idea of changing the way individuals are classified and described in the original ontology. This category was utilized in test cases 41-60.

Table 6. Results for the OAEI IIMB track for the small (large) dataset

Transformations	0-20	21-40	41-60	61-80	overall
Precision	0.99 (0.98)	0.95 (0.94)	0.96 (0.99)	0.86 (0.86)	0.94 (0.95)
Recall	0.93 (0.87)	0.83 (0.79)	0.97 (0.99)	0.54 (0.53)	0.83 (0.80)
F_1-value	0.96 (0.91)	0.88 (0.85)	0.97 (0.99)	0.65 (0.63)	0.87 (0.85)

– *Combination* This fourth set is obtained by combining together the three kinds of transformations and constitute the last test cases 61-80 in IIMB.

Data evaluation techniques. Finally, in the *data evaluation* activity, we automatically create a ground-truth in form of a reference alignment for each test case. A reference alignment contains the correct correspondences (in some contexts called "links") between the individuals in the reference ABox and the corresponding transformed individuals in the test case. These mappings are what an instance matching application is expected to find between the original ABox and the test case.

Results. The results of our approach on the IIMB 2010 benchmark are summarized in Table 6. The first numbers are the results of the small IIMB dataset containing 363 individuals, while the numbers in brackets represent our results for the large IIMB benchmark consisting of 1416 individuals. When examining the differences between the small and the large dataset, we notice that the values are slightly better for the small dataset. The F_1-values for the first category of the large dataset decrease by 0.05 compared to the small one, for the second category the disparity is 0.03, respectively. The third and forth category both have 0.02 lower F_1-values for the large dataset compared to the small one.

Since the large dataset is slightly more challenging, we report the results compared to other matching systems over the large version. Figures 5 and 6 illustrate the results for all of the participating matching systems at OAEI. Our object reconciliation approach has been implemented in the combinatorial optimization for data integration (CODI) system [36]. Besides our CODI matching application, the systems ASMOV [22] and RiMOM [52] participated in this particular track of the OAEI. ASMOV uses a weighted average of measurements of similarity along different features of ontologies, and obtains a pre-alignment based on these measurements. It then applies a process of semantic verification to reduce the amount of semantic inconsistencies. RiMOM implements several different matching strategies which are defined based on different ontological information. For each individual matching task, RiMOM can automatically and dynamically combine multiple strategies to generate a composed matching result.

Figure 5 compares the matching results with respect to precision, recall, and F_1-value. In the first category (data transformation) the ASMOV and the RiMOM system having F_1-values of 0.98 and 1.00 outperformed CODI's F_1-value of 0.91. The reason for CODI's worse performance in this category is due to the naïve lexical similarity measures CODI applies as shown in Algorithm 1.

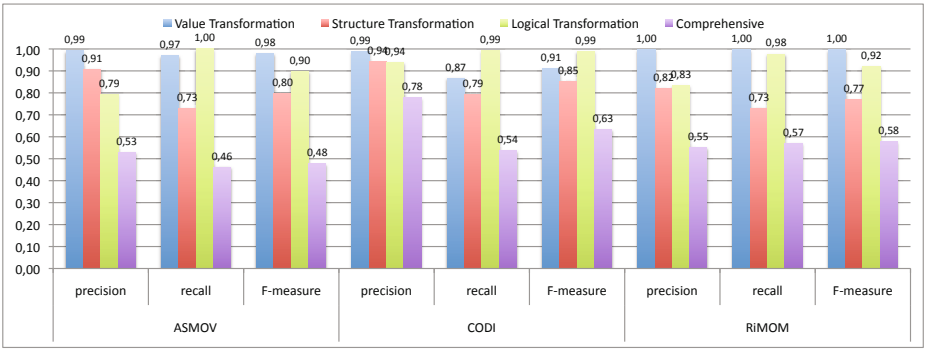


Fig. 5. Results for the large IIMB subtrack of the OAEI 2010

However, leveraging terminological structure for instance matching with Markov logic, like described in Section 5, leads to a significant improvement of CODI in the structure transformation category and the semantic transformation category. Our results compared to the ones of the ASMOV system are 5 per-cent higher in F_1-value for the structure transformation category and 9 per-cent in the semantic transformation category, respectively. The RiMOM system has 7 per-cent lower F_1-values in both the structure and the transformation category. In the last and most challenging category where all three transformation categories are combined, CODI achieved a F_1-value of 0.63 outperforming RiMOM (0.58) and ASMOV (0.48).

The precision and recall diagram in Figure 6 shows the aggregated values for recall on the x-axis and precision on the y-axis. For recall values ranging from 0.0 up to 0.6 the CODI system has the highest precision values compared to the ASMOV and RiMOM system. Only for recall values of 0.7 and higher, first the precision values of RiMOM (for recall values between 0.7 and 0.9) and then the precision values of ASMOV (for recall value 1.0) are higher.

Aggregated over all 80 test cases CODI reaches an F_1-value of 0.87 which is 5 per-cent higher than the result of ASMOV (F_1-value of 0.82) and 3 per-cent higher than RiMOM (F_1-value of 0.84)[7]. In summary, it is evident that utilizing the probabilistic-logical framework based on Markov logic for object reconciliation outperforms state-of-the-art instance matching systems.

6 Related Work

There have been a number of approaches for extending description logics with probabilistic information in the earlier days of description logics. Heinsohn [18] was one of the first to propose a probabilistic notion of subsumption for the logic ALC. Jaeger [21] investigated some general problems connected with the

[7] We refer the reader to `http://www.instancematching.org/oaei/imei2010/iimbl.html` for detailed results of every single test case and their aggregation.

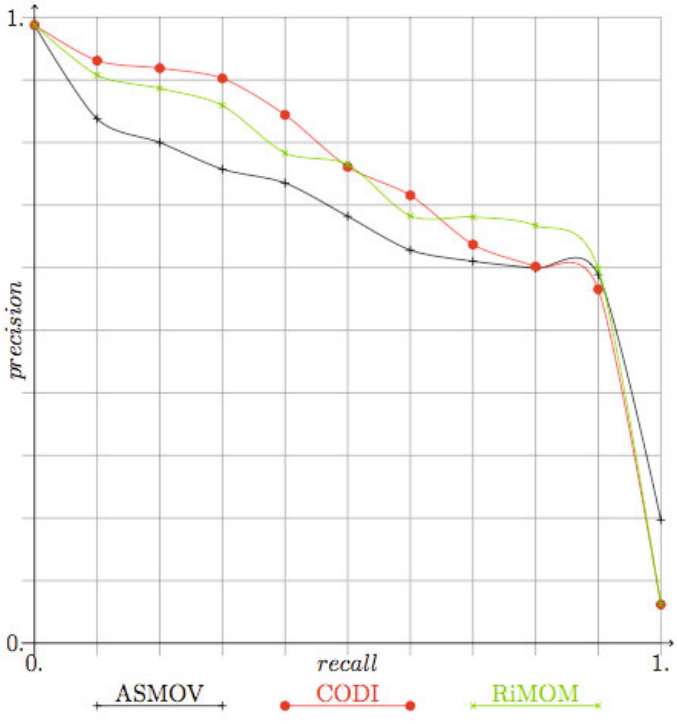


Fig. 6. Precision/recall of tools participating in the IIMB subtrack

extension of T-Boxes and ABoxes with objective and subjective probabilities and proposed a general method for reasoning with probabilistic information in terms of probability intervals attached to description logic axioms. Recently, Giugno and Lukasiewicz proposed a probabilistic extension of the logic SHOQ along the lines sketched by Jaeger [17]. A major advantage of this approach is the integrated treatment of probabilistic information about Conceptual and Instance knowledge based on the use of nominals in terminological axioms that can be used to model uncertain information about instances and relations. An alternative way of combining description logics with probabilistic information has been proposed by Koller et al. [24]. In contrast to the approaches mentioned above, the P-CLASSIC approach is not based on probability intervals. Instead it uses a complete specification of the probability distribution in terms of a Bayesian network which nodes correspond to concept expressions in the CLASSIC description logic. Bayesian networks have also been used in connection with less expressive logics such as TDL [55]. The approaches for encoding probabilities in concept hierarchies using Bayesian networks described in the section preliminaries and background can be seen as a simple special case of these approaches.

More recently proposals for combining the web ontology language OWL with probabilistic information have been proposed. The first kind of approach implements a loose coupling of the underlying semantics of OWL and probabilistic models. In particular these approaches use OWL as a language for talking about probabilistic models. An example of this approach is the work of Yang and Calmet that propose a minimal OWL ontology for representing random variables and dependencies between random variables with the corresponding conditional probabilities [54]. This allows the user to write down probabilistic models that correspond to Bayesian networks as instances of the OntoBayes Ontology. The encoding of the model in OWL makes it possible to explicitly link random variables to elements of an OWL ontology, a tighter integration on the formal level, however, is missing. A similar approach is proposed by Costa and Laskey. They propose the PR-OWL model which is an OWL ontology for describing first order probabilistic models [5]. More specifically, the corresponding ontology models Multi-Entity Bayesian networks [25] that define probability distributions over first-order theories in a modular way. Similar to OntoBayes, there is no formal integration of the two representation paradigms as OWL is used for encoding the general structure of Multi-entity Bayesian networks on the meta-level. The second kind of approaches actually aims at enriching OWL ontologies with probabilistic information to support uncertain reasoning inside OWL ontologies. These approaches are comparable with the work on probabilistic extensions of description logics also presented in this section. A survey of the existing work reveals, however, that approaches that directly address OWL as an ontology language are less ambitious with respect to combining logical and probabilistic semantics that the work in the DL area. An example is the work of Holi and Hyvonnen [19] that describe a framework for representing uncertainty in simple classification hierarchies using Bayesian networks. A slightly more expressive approach called BayesOWL is proposed by Ding and others [9]. They also consider Boolean operators as well as disjointness and equivalence of OWL classes and present an approach for constructing a Bayesian network from class expressions over these constructs. An interesting feature of BayesOWL is some existing work on learning and representing uncertain alignments between different BayesOWL ontologies reported in [38]. An additional family of probabilistic logics are log-linear description logics [35] which integrate lightweight description logics and probabilistic log-linear models.

Probabilistic approaches to ontology matching based on undirected probabilistic graphical models have recently produced competitive matching results [1]. There are numerous other non-probabilistic approaches to ontology matching and to mention all of them would be beyond the scope of this article. We refer the reader to the systems participating in the OAEI [13] which are described in the respective papers. More prominent systems with a long history of OAEI participation are Falcon [20], Aroma [8], ASMOV [23], and AgreementMaker [6].

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 [56] and OKKAM [46]. Both systems particpated in the instance matching track of the Ontology Alignment Evaluation in 2009. 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 [56]. An object reconciliation method applicable to our setting was proposed in [42] 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.

7 Conclusion

We introduced a declarative framework for web data integration based on Markov logic capturing a wide range of matching strategies. Since these strategies are expressed with a unified syntax and semantics we can isolate variations and empirically evaluate their impact. While we focused only on a small subset of possible alignment strategies the results are already quite promising. We have also successfully learned weights for soft formulas within the framework. In cases where training data is not available, weights set manually by experts still result in improved alignment quality.

We have demonstrated that both ontology matching and object reconciliation problems can be expressed in the framework. Due to the declarative nature of the approach numerous algorithms can be applied to compute the final alignments. Based on our experience, however, integer linear programming in combination with cutting plane inference and delayed column generation strategies are especially suitable since they guarantee that the hard formulas are not violated. The framework allows one to combine lexical a-priori similarities between matchable entities with the terminological knowledge encoded in the ontology. We argued that most state-of-the-art approaches for ontology and 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 alignments.

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

References

1. Albagli, S., Ben-Eliyahu-Zohary, R., Shimony, S.E.: Markov network based ontology matching. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1884–1889 (2009)
2. Bechhofer, S., Horrocks, I., Turi, D.: The OWL instance store: System description. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS (LNAI), vol. 3632, pp. 177–181. Springer, Heidelberg (2005)
3. Bhattacharya, I., Getoor, L.: Entity resolution in graphs. In: Mining Graph Data, Wiley, Chichester (2006)
4. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82(1-2), 353–367 (1996)
5. Costa, P.C.G., Laskey, K.B.: Pr-owl: A framework for probabilistic ontologies. In: Bennett, B., Fellbaum, C. (eds.) Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS). Frontiers in Artificial Intelligence and Applications, pp. 237–249. IOS Press, Amsterdam (2006)
6. Cruz, I.F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Antonelli, F.P., Keles, U.C.: Using AgreementMaker to Align Ontologies for OAEI 2010. In: Proceedings of the 5th Workshop on Ontology Matching (2010)
7. Cruz, I., Palandri, F., Antonelli, Stroe, C.: Efficient selection of mappings and automatic quality-driven combination of matching methods. In: Proceedings of the ISWC 2009 Workshop on Ontology Matching (2009)
8. David, J., Guillet, F., Briand, H.: Matching directories and OWL ontologies with AROMA. In: Proceedings of the 15th Conference on Information and knowledge management (2006)
9. Ding, L., Kolari, P., Ding, Z., Avancha, S.: Bayesowl: Uncertainty modeling in semantic web ontologies. In: Ma, Z. (ed.) Soft Computing in Ontologies and Semantic Web, Springer, Heidelberg (2006)
10. Ding, L., Finin, T.W.: Characterizing the semantic web on the web. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 242–257. Springer, Heidelberg (2006)
11. Euzenat, J., Hollink, A.F.L., Joslyn, C., Malaisé, V., Meilicke, C., Pane, A.N.J., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., dos Santos, C.T., Vouros, G.: Results of the ontology alignment evaluation initiative 2009. In: Proceedings of the ISWC 2009 workshop on Ontology Matching (2009)

12. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
13. Euzenat, J., et al.: First Results of the Ontology Alignment Evaluation Initiative 2010. In: Proceedings of the 5th Workshop on Ontology Matching (2010)
14. Fellegi, I., Sunter, A.: A theory for record linkage. Journal of the American Statistical Association 64(328), 1183–1210 (1969)
15. Ferrara, A., Lorusso, D., Montanelli, S., Varese, G.: Towards a Benchmark for Instance Matching. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, Springer, Heidelberg (2008)
16. Ferrara, A., Montanelli, S., Noessner, J., Stuckenschmidt, H.: Benchmarking Matching Applications on the Semantic Web. In: The Semantic Web: Research and Applications (2011)
17. Giugno, R., Lukasiewicz, T.: P-$\mathcal{SHOQ}(\mathbf{D})$: A probabilistic extension of $\mathcal{SHOQ}(\mathbf{D})$ for probabilistic ontologies in the semantic web. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, p. 86. Springer, Heidelberg (2002)
18. Heinsohn, J.: A hybrid approach for modeling uncertainty in terminological logics. In: Kruse, R., Siegel, P. (eds.) ECSQAU 1991 and ECSQARU 1991. LNCS, vol. 548, pp. 198–205. Springer, Heidelberg (1991)
19. Holi, M., Hyvönen, E.: Modeling uncertainty in semantic web taxonomies. In: Ma, Z. (ed.) Soft Computing in Ontologies and Semantic Web, Springer, Heidelberg (2006)
20. Hu, W., Chen, J., Cheng, G., Qu, Y.: ObjectCoref & Falcon-AO: Results for OAEI 2010. In: Proceedings of the 5th International Ontology Matching Workshop (2010)
21. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Doyle, J., Sandewall, E., Torasso, P. (eds.) Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning, pp. 305–316. Morgan Kaufmann, San Francisco (1994)
22. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: ASMOV: Results for OAEI 2010. Ontology Matching, 126 (2010)
23. Jean-Marya, Y.R., Patrick Shironoshitaa, E., Kabuka, M.R.: Ontology matching with semantic verification. Web Semantics 7(3) (2009)
24. Koller, D., Levy, A., Pfeffer, A.: P-classic: A tractable probabilistic description logic. In: Proceedings of the 14th AAAI Conference on Artificial Intelligence (AAAI 1997), pp. 390–397 (1997)
25. Laskey, K.B., Costa, P.C.G.: Of klingons and starships: Bayesian logic for the 23rd century. In: Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, pp. 346–353. AUAI Press (2005)
26. Levenshtein, V.I.: Binary codes capable of correcting deletions and insertions and reversals. In: Doklady Akademii Nauk SSSR, pp. 845–848 (1965)
27. Li, L., Horrocks, I.: A software framework for matchmaking based on semantic web technology. International Journal of Electronic Commerce 8(4), 39 (2004)
28. Meilicke, C., Stuckenschmidt, H.: Analyzing mapping extraction approaches. In: Proceedings of the Workshop on Ontology Matching, Busan, Korea (2007)
29. Meilicke, C., Stuckenschmidt, H.: An efficient method for computing alignment diagnoses. In: Proceedings of the International Conference on Web Reasoning and Rule Systems, Chantilly, Virginia, USA, pp. 182–196 (2009)
30. Meilicke, C., Tamilin, A., Stuckenschmidt, H.: Repairing ontology mappings. In: Proceedings of the Conference on Artificial Intelligence, Vancouver, Canada, pp. 1408–1413 (2007)

31. Melnik, S., Garcia-Molina, H., Rahm., E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Proceedings of ICDE, pp. 117–128 (2002)
32. Meza-Ruiz, I., Riedel, S.: Multilingual semantic role labelling with markov logic. In: Proceedings of the Conference on Computational Natural Language Learning, pp. 85–90 (2009)
33. Niepert, M.: A Delayed Column Generation Strategy for Exact k-Bounded MAP Inference in Markov Logic Networks. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (2010)
34. Niepert, M., Meilicke, C., Stuckenschmidt, H.: A Probabilistic-Logical Framework for Ontology Matching. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (2010)
35. Niepert, M., Noessner, J., Stuckenschmidt, H.: Log-Linear Description Logics. In: Proceedings of the International Joint Conference on Artificial Intelligence (2011)
36. Noessner, J., Niepert, M.: CODI: Combinatorial Optimization for Data Integration–Results for OAEI 2010. In: Proceedings of the 5th Workshop on Ontology Matching (2010)
37. Noessner, J., Niepert, M., Meilicke, C., Stuckenschmidt, H.: Leveraging Terminological Structure for Object Reconciliation. In: The Semantic Web: Research and Applications, pp. 334–348 (2010)
38. Pan, R., Ding, Z., Yu, Y., Peng, Y.: A bayesian network approach to ontology mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 563–577. Springer, Heidelberg (2005)
39. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2) (2006)
40. Riedel, S.: Improving the accuracy and efficiency of map inference for markov logic. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (2008)
41. Roth, D., Yih, W.-t.: Integer linear programming inference for conditional random fields. In: Proceedings of ICML, pp. 736–743 (2005)
42. Saïs, F., Pernelle, N., Rousset, M.-C.: Combining a logical and a numerical method for data reconciliation. Journal on Data Semantics 12, 66–94 (2009)
43. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, Chichester (1998)
44. Shavlik, J., Natarajan, S.: Speeding up inference in markov logic networks by preprocessing to reduce the size of the resulting grounded network. In: Proceedings of the 21st International Joint Conference on Artifical intelligence, pp. 1951–1956 (2009)
45. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a practical OWL-DL reasoner. Journal of Web Semantics 5(2), 51–53 (2007)
46. Stoermer, H., Rassadko, N.: 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)
47. Stuckenschmidt, H., van Harmelen, F.: Information Sharing on the Semantic Web. Advanced Information and Knowledge Processing. Springer, Heidelberg (2005)
48. Stuckenschmidt, H.: A Semantic Similarity Measure for Ontology-Based Information. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 406–417. Springer, Heidelberg (2009)
49. Svab, O., Svatek, V., Berka, P., Rak, D., Tomasek, P.: Ontofarm: Towards an experimental collection of parallel ontologies. In: Poster Track of ISWC, Galway, Ireland (2005)

50. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: Proceedings of ICML, pp. 896–903 (2005)
51. Tsarkov, D., Riazanov, A., Bechhofer, S., Horrocks, I.: Using vampire to reason with OWL. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 471–485. Springer, Heidelberg (2004)
52. Wang, Z., Zhang, X., Hou, L., Zhao, Y., Li, J., Qi, Y., Tang, J.: RiMOM Results for OAEI 2010. Ontology Matching, 195 (2010)
53. Wu, F., Weld, D.S.: Automatically refining the wikipedia infobox ontology. In: Proceeding of the International World Wide Web Conference, pp. 635–644 (2008)
54. Yang, Y., Calmet, J.: Ontobayes: An ontology-driven uncertainty model. In: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC 2005), pp. 457–463 (2005)
55. Yelland, P.M.: An alternative combination of bayesian networks and description logics. In: Cohn, A., Giunchiglia, F., Selman, B. (eds.) Proceedings of of the 7th International Conference on Knowledge Representation (KR 2000), pp. 225–234. Morgan Kaufman, San Francisco (2002)
56. Zhang, X., Zhong, Q., Shi, F., Li, J., Tang, J.: RiMOM results for OAEI 2009. In: Proceedings of the ISWC 2009 workshop on ontology matching (2009)