Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

A Probabilistic-Logical Framework for Ontology Matching

Mathias Niepert, Christian Meilicke, Heiner Stuckenschmidt
KR & KM Research Group, Universität Mannheim, Germany
{mathias, christian, heiner}@informatik.uni-mannheim.de

Abstract

Ontology matching is the problem of determining correspondences between concepts, properties, and individuals of different heterogeneous ontologies. With this paper we present a novel probabilistic-logical framework for ontology matching based on Markov logic. We define the syntax and semantics and provide a formalization of the ontology matching problem within the framework. The 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.

Introduction

Ontology matching, or ontology alignment, is the problem of determining correspondences between concepts, properties, and individuals of two or more different formal ontologies (Euzenat and Shvaiko 2007). The alignment of ontologies enables the knowledge and data expressed in the matched ontologies to interoperate. A major insight of the ontology alignment evaluation initiative (OAEI) (Jérôme Euzenat et al 2009) is that there is no best method or system for 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 result justifies not only the OAEI itself but also the need for a framework that facilitates the comparison of different strategies in a straight-forward and transparent manner. To ensure comparability of different matching approaches such a framework would need a number of characteristics. In particular it should feature

- a unified syntax that supports the specification of different approaches in the same language to isolate meaningful methodological variations and ensure that only the effects of known variations are observed;
- a well-defined semantics that guarantees that matching conditions are interpreted uniformly and that outcome variations are not merely a result of different implementations of identical features;
- a testbed for a wide range of techniques used in ontology matching including the use of soft and hard evidence such as string similarities (soft) and logical consistency of the result (hard); and
- support for the experimental comparison and standardized evaluation of techniques on existing benchmarks.

Based on these considerations, we argue that Markov logic (Richardson and Domingos 2006) 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 formulae. This allows the inclusion of both *known* logical statements and *uncertain* formulae modeling potential correspondences and structural properties of the ontologies. For instance, hard formulae can reduce incoherence during the alignment process while soft formulae can factor in a-priori confidence values for correspondences. 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 (Wu and Weld 2008) and multilingual semantic role labeling (Meza-Ruiz and Riedel 2009). Furthermore, probabilistic approaches to ontology matching have recently produced competitive matching results (Albagli, Ben-Eliyahu-Zohary, and Shimony 2009).

In this paper, we present a framework for ontology matching based on the syntax and semantics of Markov logic, in the spirit of a tool-box, allowing users to specify and combine different individual matching strategies. In particular

- we describe how several typical matching approaches are captured by the framework;
- we show how these approaches can be aggregated in a modular manner, jointly increasing the quality of the alignments; and
- we compare our framework to state-of-the-art matching systems and verify empirically that the combination of three matching strategies leads to alignments that are more accurate than those generated by any of the monolithic matching systems.

The paper is structured as follows. First, we briefly define ontology matching and introduce a running example that is used throughout the paper. We then introduce the syntax and semantics of the ML framework and show that it can represent numerous different matching approaches. 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 OAEI benchmark datasets. We conclude with a set of insights gained from the experiments and some ideas for future research.

Ontology Matching

Ontology matching is the process of detecting links between entities in heterogeneous ontologies. Based on a definition by Euzenat and Shvaiko (Euzenat and Shvaiko 2007), 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 $\langle e_1, e_2, r\rangle$ 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 following 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 throughout the paper.

Example 2. *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.*

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

If we apply a similarity measure σ based on the Levenshtein distance (Levenshtein 1965) there are four pairs of entities such that $\sigma(e_1, e_2) > 0.5$.

$$\sigma(Document, Documents) = 0.88 \quad (1)$$

$$\sigma(Reviewer, Review) = 0.75 \quad (2)$$

$$\sigma(hasWritten, writtenBy) = 0.7 \quad (3)$$

$$\sigma(PaperReview, Review) = 0.54 \quad (4)$$

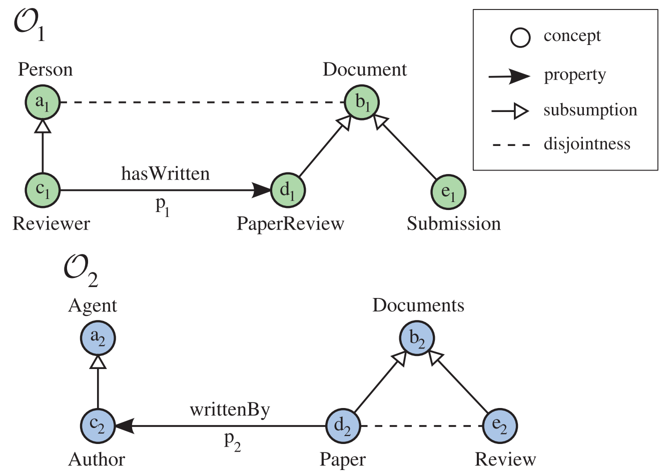

Figure 1: Example ontology fragments.

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

Markov Logic and Ontology Matching

Markov logic combines first-order logic and undirected probabilistic graphical models (Richardson and Domingos 2006). A Markov logic network (MLN) is a set of first-order formulae with weights. The more evidence we have that a formula is true the higher the weight of this formula. It has been proposed as a possible approach to several problems occurring in the context of the semantic web (Domingos et al. 2008). We argue that Markov logic provides an excellent framework for ontology matching as it captures both *hard* logical axioms and *soft* uncertain statements about potential correspondences between ontological entities. The probabilistic-logical framework we propose for ontology matching essentially adapts the syntax and semantics of Markov logic. However, we always *type* predicates and we require a strict distinction between *hard* and *soft* formulae as well as *hidden* and *observable* predicates.

Syntax

A signature is a 4-tuple $S = (O, H, C, U)$ with O a finite set of typed observable predicate symbols, H a finite set of typed hidden predicate symbols, C a finite set of typed constants, and U a finite set of function symbols. In the context of ontology matching, constants correspond to ontological entities such as concepts and properties, and predicates model relationships between these entities such as disjointness, subsumption, and equivalence. A Markov logic network (MLN) is a pair $(\mathcal{F}^h, \mathcal{F}^s)$ where $\mathcal{F}^h$ is a set $\{F_i^h\}$ of first-order formulae built using predicates from $O \cup H$ and $\mathcal{F}^s$ is a set of pairs $\{(F_i, w_i)\}$ with each F_i being a 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. Note how we explicitly distinguish between hard formulae $\mathcal{F}^h$ and soft formulae $\mathcal{F}^s$.

Semantics

Let $M = (\mathcal{F}^h, \mathcal{F}^s)$ be a Markov logic network with signature $S = (O, H, C, U)$. A *grounding* of a first-order formula F is generated by substituting each occurrence of every variable in F with constants in C of compatible type. Existentially quantified formulae are substituted by the disjunctions of their groundings over the finite set of constants. A formula that does not contain any variables is *ground* and a formula that consists of a single predicate is an *atom*. 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 (Richardson and Domingos 2006). For the ML framework, we only consider formulae with universal quantifiers at the outermost level. 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

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

if for all $F \in \mathcal{F}^h : W \models \mathcal{G}_F^C$; and $p(W) = 0$ otherwise. Here, Z is a normalization constant.

In the context of ontology matching, possible worlds correspond to possible alignments and the goal is to determine the most probable alignment given the evidence. Note that several existing methods have sought to maximize the sum of confidence values subject to constraints enforcing the alignments to be, for instance, one-to-one and functional. The given probabilistic semantics unifies these approaches in a coherent theoretical framework.

Matching Formalization

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[1]:

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

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 m_c and m_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 formulae are added to $\mathcal{F}^s$:

$$(m_c(c, d),\ \ \sigma(C, D)) \qquad \text{if C and D are concepts}$$
$$(m_p(p, r),\ \ \sigma(P, R)) \qquad \text{if P and R are properties}$$

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 (Cruz et al. 2009). 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.

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

Analogously, the same formulae can be included with hidden predicates m_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 (Meilicke, Tamilin, and Stuckenschmidt 2007). 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 formulae of the following type to $\mathcal{F}^h$.

$$dis_1(x, x') \wedge sub_2(x, x') \Rightarrow \neg(m_c(x, y) \wedge m_c(x', y'))$$
$$dis_1^d(x, x') \wedge sub_2^d(y, y') \Rightarrow \neg(m_p(x, y) \wedge m_c(x', y'))$$

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 (Meilicke and Stuckenschmidt 2009).

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

[1] Due to space considerations the list is incomplete. For instance, predicates modeling range restrictions are not included.

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* (Melnik, Garcia-Molina, and Rahm. 2002). As a reciprocal idea, the general notion of stability was introduced, expressing that an alignment should not introduce new structural knowledge (Meilicke and Stuckenschmidt 2007). 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.

$$(sub_1(x, x') \wedge \neg sub_2(y, y') \Rightarrow m_c(x, y) \wedge m_c(x', y'),\ w_1)$$
$$(sub_1^d(x, x') \wedge \neg sub_2^d(y, y') \Rightarrow m_p(x, y) \wedge m_c(x', y'),\ w_2)$$

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

The presented list of cardinality, coherence, and stability constraints is by no means exhaustive. Other constraints could, for example, model known correct correspondences or generalize the one-to-one alignment to m-to-n alignments, or 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 3. *We apply the previous formalization to Example 2. 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 2, the ground MLN would include the following relevant ground formulae. We use the concept and property labels from Figure 1 and omit ground atoms of observable predicates.*
A-priori similarity:

$$(m_c(b_1, b_2), 0.88), (m_c(c_1, e_2), 0.75),$$
$$(m_p(p_1, p_2), 0.7), (m_c(d_1, e_2), 0.54)$$

Cardinality constraints:

$$m_c(c_1, e_2) \wedge m_c(d_1, e_2) \Rightarrow c_1 = d_1 \quad (5)$$

Coherence constraints:

$$dis_1^d(p_1, b_1) \wedge sub_2^d(p_2, b_2) \Rightarrow \neg(m_p(p_1, p_2) \wedge m_c(b_1, b_2)) \quad (6)$$
$$dis_1(b_1, c_1) \wedge sub_2(b_2, e_2) \Rightarrow \neg(m_c(b_1, b_2) \wedge m_c(c_1, e_2)) \quad (7)$$
$$sub_1^d(p_1, c_1) \wedge dis_2^d(p_2, e_2) \Rightarrow \neg(m_p(p_1, p_2) \wedge m_c(c_1, e_2)) \quad (8)$$

MAP Inference as Alignment Process

Hidden predicates model correspondences between entities of the two ontologies whereas observable ones model predicates occurring in typical description logic statements. If we want to determine the most likely alignment of two given ontologies, 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 the ground formulae of $\mathcal{F}^h$ and $\mathcal{F}^s$. 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. 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 alignment we have to compute

$$\operatorname*{argmax}_{\mathbf{H}' \subseteq \mathbf{H}} \sum_{(F_i, w_i) \in \mathcal{F}^s} \sum_{g \in \mathcal{G}^C_{F_i}:\ \mathbf{O}' \cup \mathbf{H}' \models g} w_i,$$

subject to $\mathbf{O}' \cup \mathbf{H}' \models \mathcal{G}^C_F$ for all $F \in \mathcal{F}^h$.
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. Several approximate algorithms for the weighted MAX-SAT problem exist. However, since each ground formula in $\mathcal{F}^h$ must be satisfied in the computed MAP state, exact inference is required in our setting. Hence, we apply integer linear programming (ILP) which was shown to be an effective method for exact MAP inference in undirected graphical models (Roth and Yih 2005; Taskar et al. 2005) and specifically in Markov logic networks (Riedel 2008). ILP is concerned with optimizing a linear objective function over a finite number of integer variables, subject to a set of linear equalities and inequalities over these variables (Schrijver 1998). We omit the details of the ILP representation of a ground Markov logic network but demonstrate how the ground formulae from Example 3 would be represented as an ILP instance.

Example 4. *Let the binary ILP variables x_1, x_2, x_3, and x_4 model the ground atoms $m_c(b_1, b_2), m_c(c_1, e_2), m_p(p_1, p_2)$, and $m_c(d_1, e_2)$, respectively. The ground formulae from Example 3 can be encoded with the following ILP.*
Maximize: $0.88x_1 + 0.75x_2 + 0.7x_3 + 0.54x_4$
Subject to:

$$x_2 + x_4 \leq 1 \quad (9)$$
$$x_1 + x_3 \leq 1 \quad (10)$$
$$x_1 + x_2 \leq 1 \quad (11)$$
$$x_2 + x_3 \leq 1 \quad (12)$$

The a-priori weights 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 formulae (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 $\{m_c(b_1, b_2), m_c(c_1, e_2), m_p(p_1, p_2)\}$ which would be the outcome without coherence constraints.

Experiments

We use the Ontofarm dataset (Svab et al. 2005) 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 (Jérôme Euzenat et al 2009). 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 Levenshtein 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 (Sirin et al. 2007) to create the ground MLN formulation and used TheBeast[2] (Riedel 2008) to convert the MLN formulations to the corresponding ILP instances. Finally, we applied the mixed integer programming solver SCIP[3] 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 F1-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:

- **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 (Ding and Finin 2006). 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 formulae. 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 F1-value for the remaining pair of ontologies. This was repeated for each of the 21 possible combinations to determine the mean of the F1-values.

[2] http://code.google.com/p/thebeast/

[3] http://scip.zib.de/

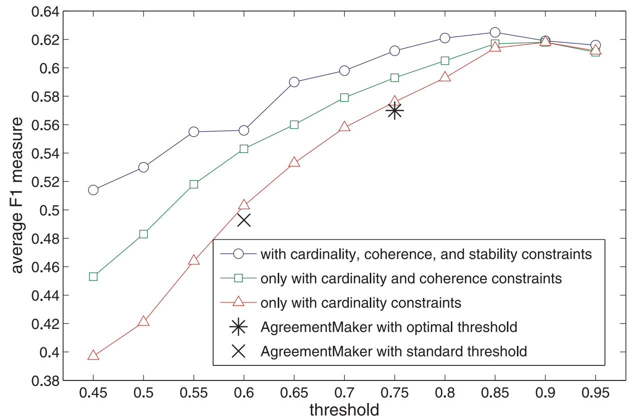

Figure 2: F1-values for **ca**, **ca+co**, and **ca+co+sm** averaged over the 21 OAEI reference alignments for thresholds ranging from 0.45 to 0.95. AgreementMaker was the best performing system on the conference dataset of the latest ontology evaluation initiative in 2009.

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

Table 1: Average F1-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.

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 2 depicts the average F1-values for **ca**, **ca+co**, and **ca+co+sm** compared to the average F1-values achieved by AgreementMaker (Cruz et al. 2009), the best-performing system in the OAEI conference track of 2009. These average F1-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 F1-value attains its maximum. The inclusion of coherence constraints (**ca+co**) improves the average F1-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 F1-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 F1-values > 0.57 for $\tau \geq 0.75$ the **ca+co+sm** setting generates F1-values > 0.59 for $\tau \geq 0.65$. Even for $\tau = 0.45$, usually considered an inappropriate threshold choice, we measured an average F1-value of 0.51 and average precision and recall values of 0.48 and 0.60, respectively. Table 1 compares the average F1-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.

Discussion and Future Work

We presented a Markov logic based framework for ontology matching 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 effects. Even though 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 formulae within the framework. In cases where training data is not available, weights set manually by experts still result in improved alignment quality. Research related to determining appropriate weights based on structural properties of ontologies is a topic of future work. The framework is not only useful for aligning concepts and properties but can also include instance matching. For this purpose, one would only need to add a hidden predicate modeling instance correspondences. The resulting matching approach would immediately benefit from probabilistic joint inference, taking into account the interdependencies between terminological and instance correspondences.

Acknowledgments

Many thanks to Sebastian Riedel for helping us tame TheBeast and for his valuable feedback.

References

Albagli, S.; Ben-Eliyahu-Zohary, R.; and Shimony, S. E. 2009. Markov network based ontology matching. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1884–1889.

Cruz, I.; Palandri, F.; Antonelli; and Stroe, C. 2009. Efficient selection of mappings and automatic quality-driven combination of matching methods. In *Proceedings of the ISWC 2009 Workshop on Ontology Matching*.

Ding, L., and Finin, T. 2006. Characterizing the semantic web on the web. In *Proceedings of the International Semantic Web Conference 2006*, 242–257.

Domingos, P.; Lowd, D.; Kok, S.; Poon, H.; Richardson, M.; and Singla, P. 2008. Just add weights: Markov logic for the semantic web. In *Proceedings of the Workshop on Uncertain Reasoning for the Semantic Web*, 1–25.

Euzenat, J., and Shvaiko, P. 2007. *Ontology matching*. Springer-Verlag.

Jérôme Euzenat et al. 2009. Results of the ontology alignment evaluation initiative 2009. In *Proceedings of the Workshop on Ontology Matching*.

Levenshtein, V. I. 1965. Binary codes capable of correcting deletions and insertions and reversals. *Doklady Akademii Nauk SSSR* 845–848.

Meilicke, C., and Stuckenschmidt, H. 2007. Analyzing mapping extraction approaches. In *Proceedings of the Workshop on Ontology Matching*.

Meilicke, C., and Stuckenschmidt, H. 2009. An efficient method for computing alignment diagnoses. In *Proceedings of the International Conference on Web Reasoning and Rule Systems*, 182–196.

Meilicke, C.; Tamilin, A.; and Stuckenschmidt, H. 2007. Repairing ontology mappings. In *Proceedings of the Conference on Artificial Intelligence*, 1408–1413.

Melnik, S.; Garcia-Molina, H.; and Rahm., E. 2002. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of ICDE*, 117–128.

Meza-Ruiz, I., and Riedel, S. 2009. Multilingual semantic role labelling with markov logic. In *Proceedings of the Conference on Computational Natural Language Learning*, 85–90.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Riedel, S. 2008. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of UAI*, 468–475.

Roth, D., and Yih, W. 2005. Integer linear programming inference for conditional random fields. In *Proceedings of ICML*, 736–743.

Schrijver, A. 1998. *Theory of Linear and Integer Programming*. Wiley & Sons.

Sirin, E.; Parsia, B.; Grau, B. C.; Kalyanpur, A.; and Katz, Y. 2007. Pellet: a practical OWL-DL reasoner. *Journal of Web Semantics* 5(2):51–53.

Svab, O.; Svatek, V.; Berka, P.; Rak, D.; and Tomasek, P. 2005. Ontofarm: Towards an experimental collection of parallel ontologies. In *Poster Track of ISWC*.

Taskar, B.; Chatalbashev, V.; Koller, D.; and Guestrin, C. 2005. Learning structured prediction models: a large margin approach. In *Proceedings of ICML*, 896–903.

Wu, F., and Weld, D. S. 2008. Automatically refining the wikipedia infobox ontology. In *Proceeding of the International World Wide Web Conference*, 635–644.