A New Paradigm for Alignment Extraction

Christian Meilicke and Heiner Stuckenschmidt

Research Group Data and Web Science
University of Mannheim, 68163 Mannheim, Germany
christian|heiner@informatik.uni-mannheim.de

Abstract. Ontology matching techniques that are based on the analysis of names usually create first a set of matching hypotheses annotated with similarity weights followed by the extraction or selection of a set of correspondences. We propose to model this last step as an optimization problem. Our proposal differs fundamentally from other approaches since both logical and linguistic entities appear as first class citizens in the optimization problem. The extraction step will not only result in a set of correspondences but will also entail assumptions related to the meaning of the tokens that appeared in the involved labels. We discuss examples that illustrate the benefits of our approach and present a Markov Logic formalization. We conduct an experimental evaluation and present first results.

1 Introduction

Ontology Matching has become a vivid field of research over the last decade. Hundreds of papers propose and discuss ontology matching techniques, introduce improvements, or present complete matching systems. Especially the system papers illustrate a general paradigm common to probably all systems using name-based alignment methods. This paradigm is the understanding of ontology matching as a sequential process that starts with analyzing different types of evidence, in most cases with a focus on the involved labels, and generates as an intermediate result a set of weighted matching hypotheses. From the intermediate result a subset of the generated hypotheses is chosen as final output. The first phase is typically dominated by the computation, aggregation, propagation, and any other method for refining similarity scores. The techniques applied in the second phase range from thresholds to the selection of coherent subsets [6, 8] that might be optimal with respect to an objective function. Most approaches model the intermediate result as a set of correspondences annotated with confidence scores. These confidence scores are aggregated values derived from an analysis of the tokens that appear in the labels of the ontological entities. With the help of several examples we argue that the extraction problem should be modeled differently such that both tokens and logical entities (classes and properties) appear as first class citizens. Otherwise it will not be possible to exploit that the acceptance or rejection of a correspondence follows from the assumption that two tokens have (or do not have) the same meaning. However, any reasonable extraction should be consistent with its underlying assumptions. This can only be ensured if the assumptions themselves can be modeled explicitly.

We presented a first sketch of this approach in [9]. Now we extend and concretize the approach including a first implementation. We present foundations in Section 2. In

Section 3 we discuss two scenarios where a classic approach makes a selection decision in an non-reasonable way. In Section 4 we present our approach and explain how to deal with the issues mentioned before. Experimental results of a first prototypical implementation are presented in Section 5 before concluding in Section 6.

2 Foundations

We introduce some technical terms (Section 2.1), describe state of the art methods for extracting an alignment (Section 2.2), and take a closer look at one them (Section 2.3).

2.1 Nomenclature

Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies that have to be matched. A correspondence is a quadruple $\langle e_1, e_2, r, c\rangle$ where e and e' are entities defined in $\mathcal{O}_1$ and $\mathcal{O}_2$. r is a semantic relation between e_1 and e_2. Within this paper the semantic relation will always be equivalence and e_1 and e_2 will always be classes or (data or object) properties. The numerical value c is referred to as confidence value. The higher the value, the higher is the probability that $r(e_1, e_2)$ holds. The confidence value is an optional element and will sometimes be omitted. The outcome of a matching system is a set of correspondences between $\mathcal{O}_1$ and $\mathcal{O}_2$. Such a set is called an alignment $\mathcal{A}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$.

In the following we distinguish between linguistic entities (labels and tokens) and ontological entities (classes and properties) using the following naming convention.

`n#ClassOrProperty` - Refers to a class or property in $\mathcal{O}_n$ (with $n \in 1, 2$).
`n:Label` - Refers to a label used in $\mathcal{O}_n$ as a class or property description.
`n:Token`$_t$ - Refers to a token that appears as a part of a label in $\mathcal{O}_n$.

We will later, e.g., treat `1#AcceptedPaper` and `1:AcceptedPaper` as two different entities. The first entity appears in logical axioms and the second might be a description of the first entity. The label consists of the tokens `1:Accepted`$_t$ and `1:Paper`$_t$. We need three types of entities (logical entities, labels, tokens) because a logical entity can be described by several labels and a label can be decomposed in several tokens.

2.2 Alignment Extraction

The easiest way for selecting a final alignment $\mathcal{A}$ from a set of matching hypotheses $\mathcal{H}$ is the application of a threshold. However, a threshold does not take into account any dependencies between correspondences in $\mathcal{H}$. Thus, it might happen that an entity `1#e` is mapped on `2#e'` and `2#e''` even though `2#e'` and `2#e''` are located in different branches of the concept hierarchy.

This can be solved easily. We first sort $\mathcal{H}$ by confidence scores. Starting with an empty alignment $\mathcal{A}$, we iterate over $\mathcal{H}$ and add each $\langle e_1, e_2, =, c\rangle \in \mathcal{H}$ to $\mathcal{A}$ if $\mathcal{A}$ does not yet contain a correspondence that links one of e_1 or e_2 to some other entity. This ensures that $\mathcal{A}$ is finally a one-to-one alignment. Similar algorithms can be applied to ensure that certain anti-pattern (e.g., Asmov [5]) are avoided when adding correspondences to $\mathcal{A}$. It is also possible to use reasoning to guarantee the coherence of the

generated alignment (e.g., Logmap [6]). Checking a set of patterns is then replaced by calling a reasoning engine.

Such an approach needs to decide upon the order in which correspondences are iterated over because different orders can lead to different results. Global methods try to overcome this problem. Similarity flooding [10], for example, is based on the following assumption: The similarity between two entities linked by a correspondence in $\mathcal{H}$ must depend on the similarity of their adjacent nodes for which an initial similarity is specified in $\mathcal{H}$. The algorithm does not select a subset of $\mathcal{H}$ as final outcome but generates a refined similarity distribution over $\mathcal{H}$. Other global methods explicitly define an optimization problem in which a subset from $\mathcal{H}$ needs to be chosen that maximizes an objective function. This is detailed in the following section.

2.3 Global Optimization with Markov Logic

In [13] and [2] Markov Logic has been proposed to solve the alignment extraction problem. The authors have argued that the solution to a given matching problem can be obtained by solving the maximum a-posteriori (MAP) problem of a ground Markov logic network. In such a formalization the MAP state, which is the solution of an optimization problem, corresponds to the most probable subset $\mathcal{A}$ of $\mathcal{H}$. In the following we explain the basic idea of the approach proposed in [13]. Due to the lack of space we omit a theoretical introduction to Markov Logic and refer the reader to [15].

In [13] the authors have defined, due to the fact that Markov Logic is a log linear probabilistic model, the objective function as the confidence total of $\mathcal{A} \subseteq \mathcal{H}$. Without any further constraints and given that all confidences are positive it follows that $\mathcal{A} = \mathcal{H}$. However, some of the constraints that have been mentioned above can easily be encoded as first-order formulae in Markov Logic. We can postulate that a pair of correspondences violating the 1:1 constraint is not allowed in the final solution. This can be expressed as follows.

$$map(e_1, e_2) \wedge map(e_1', e_2') \wedge e_1 = e_1' \rightarrow e_2 = e_2'$$

Similarly, coherence constraints can be added to avoid certain patterns of incoherent mappings. An example is the constraint that the classes e_1 and e_1' where e_1' is a subclass of e_1 cannot be mapped on e_2 and e_2' where e_2 and e_2' are disjoint:

$$sub(e_1, e_1') \wedge dis(e_2, e_2') \rightarrow \neg(map(e_1, e_2) \wedge map(e_1', e_2'))$$

Due to the lack of space, we cannot specify all constraints of the complete formalization. Additional constraints are required to take into account that properties can also be involved in logical inconsistencies (see [13]). Moreover, there are some soft constraints that reward homomorphism introduced by the selected correspondences.

Given such a formalization, a reasoning engine for Markov Logic can be used to compute the MAP state which corresponds to the most probable consistent mapping. In our terminology we call this mapping a global optimal solution. Note that the entities that appear in such a formalization are logical entities (classes and properties) only, while labels or token are completely ignored. The have only been used to compute weights for the matching hypotheses, which are the weights attached to the map-atoms.

3 Illustrating Examples

In Section 3.1 and 3.2 we analyze examples that illustrate problems of the classical approaches described in the previous section. In Section 3.3 we discuss the possibility to cope with these problems without introducing a new modeling style.

3.1 Multiple Token Occurrences

For most matching problems some of the tokens used in the labels will appear in more than one label. This is in particular the case for compound labels that can be decomposed into modifier and head noun. Figure 1 shows a typical example.

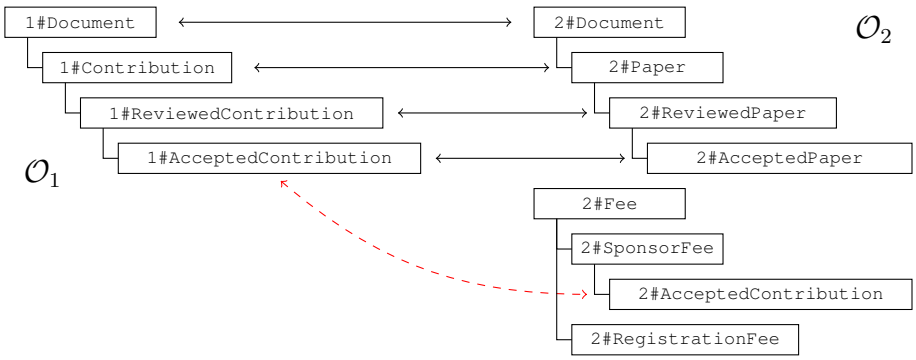

Fig. 1. Example of a non-trivial matching problem.

Let us first discuss a simplified version of the example where we ignore the branch in $\mathcal{O}_2$ rooted at the `2#Fee` class. Note that a matching problem very similar to the simplified example can be found in the OAEI conference dataset (testcase conference-ekaw). For this small excerpt there are four correspondences (solid arrows) in the reference alignment. Probably, most systems would generate $\langle$`1#Document`, `2#Document`, $=\rangle$ due to the usage of the same label. The same does not hold for the other three correspondences. For two of them the labels can be decomposed into modifier and headnoun. For all of these correspondences it is crucial to answer the question whether the words `1:Contribution` and `2:Paper` have the same meaning. How would a standard approach deal with this example? In such an approach a similarity metric would be used to compute a similarity for all relevant pairs of words. This would probably also result in a (numerical) similarity for the pair $\langle$`1:Contribution`, `2:Paper`$\rangle$, for example $sim($`1:Contribution`, `2:Paper`$) = 0.3$. This similarity would then be aggregated into a score that might result into a set of weighted hypotheses $\mathcal{H}$.

$$c_1 = \langle \texttt{1\#Document}, \texttt{2\#Document}, =, 1.0 \rangle$$
$$c_2 = \langle \texttt{1\#Contribution}, \texttt{2\#Paper}, = 0.3 \rangle$$
$$c_3 = \langle \texttt{1\#ReviewedContribution}, \texttt{2\#ReviewedPaper}, = 0.65 \rangle$$
$$c_4 = \langle \texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedPaper}, = 0.65 \rangle$$

At this stage we have lost the dependency between our final decision and the question whether or not the words `1:Contribution` and `2:Paper` have the same meaning. Without being aware of this dependency it might happen that c_1, c_3, c_4 and not c_2 are selected. This would, obviously, be an inconsistent decision, because the selection of c_3 and c_4 should always result in the selection of c_2.

One might criticize that we are making (invalid) assumptions. Above we used the average for aggregating confidences. One might also use, for example, the minimum. This results in the same confidences for c_2, c_3 and c_4. Nevertheless, the distance between `1:Contribution` = `2:Paper` is taken into account not once but several times. Thus, the decision related to c_2 will not be affected by the possibility of generating c_3 and c_4, while a human expert would take c_3 and c_4 into account.

Let us now analyze the extended example where we have the additional branch that deals with fees and (monetary) contributions. Now we have another (incorrect) matching candidate.

$$c_5 = \langle \texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedContribution}, =, 1.0 \rangle$$

Obviously, c_5 is in a 1:1 conflict with c_4. A consistent 1:1 mapping might thus consist of c_1, c_2, c_3 and c_4 or (exclusive!) c_5. However, taking the involved tokens and their possible meanings into account, we should not generate an alignment that contains c_2 and c_5 at the same time. Such an alignment will only be correct, if the tokens in $\mathcal{O}_1$ are used in an inconsistent way.

The classical approach cannot handle such cases in the appropriate way. As long as the tokens themselves are not explicitly modeled as entities in the extraction phase, unreasonable and inconsistent decisions, inconsistent with respect to assumptions related to the use of words, are made.

3.2 Ignoring Modifiers

We illustrate another pattern by an example taken from the OAEI conference dataset, namely the confof-ekaw testcase. The reference alignment for this testcase contains 20 correspondences, here we are interested in the following three correspondences.

$$\langle \texttt{1\#Banquet}, \texttt{2\#ConferenceBanquet}, = \rangle$$
$$\langle \texttt{1\#Participant}, \texttt{2\#ConferenceParticipant}, = \rangle$$
$$\langle \texttt{1\#Trip}, \texttt{2\#ConferenceTrip}, = \rangle$$

The developer of $\mathcal{O}_2$ was more verbose than the developer of $\mathcal{O}_1$. In $\mathcal{O}_2$ some of the labels have been extended by adding the prefix modifier `2:Conference`. This modifier has been omitted in $\mathcal{O}_1$ because each of the participants, trips and banquets is implicitly always associated to a conference. We are not interested in pros and cons of both styles. Both exist and a matching system should be able to cope with them.

Let us again think how we, as reasonable agents, would deal with this issue. After studying the $\mathcal{O}_1$ ontology, we would come to the decision, that it might make sense to ignore the token $\texttt{1:Conference}_t$ whenever it appears as modifier. Maybe we would first try to match both ontologies without ignoring the modifier, then we would

match both ontologies while ignoring $\texttt{1:Conference}_t$ when it appears as modifier. In both cases we ensure the coherency of the generated alignment. For our example the outcome would be that the second approach allows to generate three additional correspondences that do not introduce any logical conflicts. Thus, ignoring the modifier `1:Conference` seems to be a good choice.

Again, we can see that a first class citizen in such considerations are linguistic entities. We make certain decisions about the role of tokens and their implications result in the acceptance of correspondences, while logical constraints that deal with ontological entities have also an impact on our interpretation of tokens.

3.3 Work Around

In [12] the authors have proposed a measure called extended Tversky similarity that copes with the situation described in Section 3.2. Their idea is to weigh each token by its information content. A token like `2:Conference` that appears very often has a very low weight. It follows that a relatively high confidence score is assigned to a correspondence like $\langle \texttt{1\#Banquet}, \texttt{2\#ConferenceBanquet}, = \rangle$ because `2:Conference` has only a limited discriminative power. Note that this approach is still based on the principle to assign confidences to correspondences. Once this assignment has been made, the tokens that have been involved are no longer taken into account.

This technique has been implemented in the YAM++ matcher. This matcher achieved very good results the OAEI 2012 campaign [1] (see also the results table in Section 5). However, not the number of token-occurrences is important, but the maximal number of additional coherent correspondences that would result from ignoring a modifier. While these numbers are often correlated, this is not necessarily the case. Suppose that we have an ontology that contains the class `1#PaperAuthor` and the property `1#paperTitle`, as well as some other labels that contain the token $\texttt{1:paper}_t$. Let the other ontology contain a class `2#Author` (including authors of reviews) and a property `2#title` (to describe the title of a conference). In $\mathcal{O}_1$ we have a relatively high number of $\texttt{1:paper}_t$-token occurrences, however, the word $\texttt{1:paper}_t$ is in most cases a feature that needs to be taken into account. This can be derived from the fact that $\langle \texttt{1\#PaperAuthor}, \texttt{2\#Author}, = \rangle$ and $\langle \texttt{1\#paperTitle}, \texttt{2\#title}, = \rangle$ cannot be added without introducing logical conflicts given a meaningful axiomatization in $\mathcal{O}_1$ and $\mathcal{O}_2$. In our approach we will be able to take such cases into account.

4 Approach

We first present our approach and it formalization in Section 4.1 followed by an analysis of its impact in Section 4.2 where we revisit the examples of the previous section.

4.1 Formalization

In the following we distinguish explicitly between entities from two different layers. The first layer is the layer of labels and tokens; the entities that appear in the second layer are classes and properties. In our approach we treat entities from both layers as first

class citizens of an optimization problem. Thus, we can define the objective function of our optimization problem on top of token similarities (first layer) instead of using confidence values attached to correspondences (second layer).

Hidden predicates	
$map(e_1, e_2)$	e_1 is mapped on e_2, i.e. $\langle e_1, e_2, = \rangle \in \mathcal{A}$
$equiv_t(t_1, t_2)$	t_1 and t_2 have the same meaning
$equiv_l(l_1, l_2)$	l_1 and l_2 have the same meaning
$ignore(t)$	token t can be ignored if it appears as a modifier
Logical predicates	
$sub(e_1, e_2)$	class/property e_1 is subsumed by class/property e_2
$dis(e_1, e_2)$	e_1 and e_2 are disjoint classes
$dom(e_1, e_2)$	class e_1 is the domain of property e_2
$ran(e_1, e_2)$	class e_1 is the range of property e_2
Linguistic predicates	
$pos1(l, t)$	label l has token t at first position
$pos2(l, t)$	label l has token t at second position
$pos3(l, t)$	label l has token t at third position
$has1Token(l)$	label l is composed of one token
$has2Token(l)$	label l is composed of two tokens
$has3Token(l)$	label l is composed of three tokens
$hasLabel(e, l)$	entity e is described by label l

Table 1. Variables starting with e refer to classes or properties, e.g., `1#ConferenceFee`; l refers to complete labels, e.g., `1:ConferenceFee`, and t refers to tokens, e.g., $\texttt{1:Fee}_t$

We extend the approach described in Section 2.3, i.e., we use Markov Logic and most of the constraints presented above. However, we also need a rich set of (new) predicates listed in Table 1 to support our modeling style. The first four predicates in the listing are hidden predicates. This means that we do not know in advance if the ground atoms for these predicates are true or wrong. We attach a weight in the range $[-1.0, 0.0]$ to the atoms instantiating the $equiv_t$ predicate, if we have some evidence that the respective tokens have a similar meaning. We explicitly negate the atom if there is no such evidence. As a result we have a fragment as input that might look like this.

$$
\begin{aligned}
equiv_t(\texttt{1:Accepted}_t, \texttt{2:Accepted}_t)\ &, 0.0 \\
equiv_t(\texttt{1:Organization}_t, \texttt{2:Organisation}_t)\ &, -0.084 \\
equiv_t(\texttt{1:Paper}_t, \texttt{2:Contribution}_t)\ &, -0.9 \\
\neg equiv_t(\texttt{1:Accepted}_t, \texttt{2:Rejected}_t)\ &\ \ unweighted
\end{aligned}
$$

We do not add any (weighted or unweighted) groundings of the map, $equiv_l$, and $ignore$ predicates to the input. Our solution will finally consist of a set of atoms that are groundings of the four hidden predicates. While we are mainly interested in the map-atoms (each atom refers to a correspondence), the groundings of the other predicates can be seen as additional explanations for the finally generated alignment. These atoms

inform us which tokens and labels are assumed to be equivalent and which tokens have been ignored.

The other predicates in the table are used to describe observations relevant for the matching problem. We describe the relations between tokens and labels and the relation between labels and logical entities.

$$pos1(\texttt{1:AcceptedPaper}, \texttt{1:Accepted}_t)$$
$$pos2(\texttt{1:AcceptedPaper}, \texttt{1:Paper}_t)$$
$$has2Token(\texttt{1:AcceptedPaper})$$
$$hasLabel(\texttt{1\#AcceptedPaper, 1:AcceptedPaper})$$

We postulate that a label is matched if and only if all of its tokens are matched. We specify this explicitly for labels of different size.[1] The 2-token case is shown here.

$$has2Token(l_1) \wedge has2Token(l_2) \wedge pos1(l_1, t_{11}) \wedge pos2(l_1, t_{12}) \wedge$$
$$pos1(l_2, t_{21}) \wedge pos2(l_2, t_{22}) \rightarrow (equiv_l(l_1, l_2) \leftrightarrow equiv_t(t_{11}, t_{21}) \wedge equiv_t(t_{12}, t_{22}))$$

Next, we have to establish the connection between label and logical entity. A logical entity is matched if and only if at least one of its labels is matched.

$$map(e_1, e_2) \leftrightarrow \exists l_1 \, \exists l_2 \, (hasLabel(e_1, l_1) \wedge hasLabel(e_2, l_2) \wedge equiv_l(l_1, l_2))$$

We follow the classic approach and translate (a subset of) the ontological axioms to our formalism by using the logical predicates. We add several constraints as restrictions of the map-predicate ensuring that the generated alignment is a 1:1 mapping and that this mapping is coherent taking the ontological axioms into account. These constraints have already been explained in [13] and we can integrate them easily in our approach as constraints on the second layer. In addition to the 1:1 constraint for the map predicate, we also add a 1:1 constraint for the $equiv_t$-predicate on the token layer. This ensures that $equiv(\texttt{1:Paper}_t, \texttt{2:Contribution}_t)$ and $equiv(\texttt{1:Contribution}_t, \texttt{2:Contribution}_t)$ cannot be true at the same time.

Computing the MAP state for the modeling described so far will always yield an empty result, because the summands in the objective function are only the weights attached to the $equiv_t$-atoms. All of them are ≤ 0, thus, the best objective will be 0, which is the objective of an empty mapping. We have to add a weighted rule that rewards each correspondence, i.e., a rule that rewards each instantiation of the map predicate. We have set the reward to 0.5.

$$map(e_1, e_2), +0.5$$

Now each correspondence added to the solution increases the score of the objective by 0.5. At the same time each instantiation of the map predicate forces to instantiate at least one $equiv_l$-atom, which again forces to instantiate the related $equiv_t$-atoms weighted with values lower or equal to zero. Thus, we have defined a non trivial optimization

[1] We have not included labels with more than three tokens in our first implementation. For larger labels, we decided to match these labels directly if they are the same after normalization.

problem in which the idea of generating a comprehensive alignment conflicts with our assumptions related to the meaning of words.

Finally, we need to explain the role of the $ignore$ predicate. We want to match a 1-token label to a 2-token label if and only if we are allowed to ignore the modifier of the 2-token label and if the remaining token is equivalent to the token of the 1-token label. This can be expressed as follows.

$$has1Token(l_1) \wedge has2Token(l_2) \wedge pos1(l_1, t_{11}) \wedge pos1(l_2, t_{21}) \wedge$$
$$pos2(l_2, t_{22}) \rightarrow (equiv_l(l_1, l_2) \leftrightarrow equiv_t(t_{11}, t_{22}) \wedge ignore(t_{21}))$$

However, a modifier should not be ignored be default. For that reason we have to add again a simple weighted rule.

$$ignore(t), -0.95$$

Together, with the previous constraint this rule assigns a punishment to ignoring a token that is used as modifier. Note that the weight is set to a value lower than -0.5. By setting the value to -0.95 it will only pay off to ignore a token if it will result in at least two additional correspondences ($n \times 0.5 - 0.95 > 0.0$ for $n \geq 2$).

4.2 Impact

For the small fragment depicted in Figure 1 (from Section 3.1), we present the weighted input atoms (marked with an I) and the resulting output atoms (marked with an O) in the following listing. We omit the atoms describing the relations between tokens, labels, and logical entities, as well as those that model the logical axioms.

I	O	$equiv_t(\texttt{1:Document}_t, \texttt{2:Document}_t)$	*input weight 0.0*
I	O	$equiv_t(\texttt{1:Reviewed}_t, \texttt{2:Reviewed}_t)$	*input weight 0.0*
I	O	$equiv_t(\texttt{1:Accepted}_t, \texttt{2:Accepted}_t)$	*input weight 0.0*
I		$equiv_t(\texttt{1:Contribution}_t, \texttt{2:Contribution}_t)$	*input weight 0.0*
I	O	$equiv_t(\texttt{1:Contribution}_t, \texttt{2:Paper}_t)$	*input weight -0.9*
	O	$equiv_l(\texttt{1:Document}, \texttt{2:Document})$	
	O	$equiv_l(\texttt{1:Contribution}, \texttt{2:Paper})$	
	O	$equiv_l(\texttt{1:ReviewedContribution}, \texttt{2:ReviewedPaper})$	
	O	$equiv_l(\texttt{1:AcceptedContribution}, \texttt{2:AcceptedPaper})$	
	O	$c_1 \approx map(\texttt{1\#Document}, \texttt{2\#Document})$	
	O	$c_2 \approx map(\texttt{1\#Contribution}, \texttt{2\#Paper})$	
	O	$c_3 \approx map(\texttt{1\#ReviewedContribution}, \texttt{2\#ReviewedPaper})$	
	O	$c_4 \approx map(\texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedPaper})$	

The generated solution consists of four $equiv_t$-atom, four $equiv_l$-atoms, and four map-atoms. The four map-atoms are converted to the four correspondences of the output alignment $\{c_1, c_2, c_3, c_4\}$. The objective of this solution is $1.1 = 4 \times 0.5 + 0.0 + 0.0 + 0.0 + 0.0 - 0.9$. The example shows that the low similarity between $\texttt{1:Paper}_t$ and

$\texttt{2:Contribution}_t$ atom is compensated by the possibility to generate four correspondences. The same result would not have been achieved by attaching aggregated weights directly to the map-atoms.

Let us compare this solution to other possible and impossible solutions. Thus, let $c_5 \approx map(\texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedContribution})$ and let $c_6 \approx map(\texttt{1\#Contribution}, \texttt{2\#AcceptedContribution})$. .

$$\begin{aligned}
\text{objective for } \{c_1, c_2, c_3, c_4\} &= 4 \times 0.5 - 0.9 = 1.1 \\
\text{objective for } \{c_1, c_5\} &= 2 \times 0.5 = 1.0 \\
\{c_1, c_2, c_3, c_4, c_5\} &\quad \textit{is invalid against 1:1 constraint on the token layer} \\
\text{objective for } \{c_1\} \text{ or } \{c_5\} &= 1 \times 0.5 = 0.5 \\
\text{objective for } \{c_1, c_6\} &= 2 \times 0.5 - 0.95 = 0.05
\end{aligned}$$

The alignment $\{c_1, c_5\}$ is listed with a relatively high objective. Note that $\{c_1, c_5\}$ would be invalid, if we there would be a disjointness statement between `2#Fee` and `2#Document` due a constraint on the layer of ontological entities. We have also added $\{c_1, c_6\}$ to our listing. It illustrates the possibility to ignore a modifier. However, this solution has a low objective and there are other solutions with a better objective.

5 Preliminary Evaluation Results

In the following we report about experiments with a prototypical implementation based on the formalization presented above. The formalization is extended as follows.

- We added the constraint that if a property p is matched on a property p', then the domain (range) of p has to be matched to the domain of p' or to a direct super or subclass of the domain (range) of p'. In the latter case a small negative weight is added to the objective.
- We derived alternative labels from the directly specified labels by ignoring certain parts. For example, we added the label `1:writes` to a property labeled with `1:writesPaper`, if `1:Paper` was the label of that properties domain.
- We derived alternative labels by adding `1:ConferenceMember` as alternative label given a label like `1:MemberOfConference`.
- We added rules that allow to match two-token labels on three-token labels in case that all tokens from the two-token label are matched, however, such a case was punished with a negative weight.

We use the following basic techniques for computing the input similarity scores. First we normalize and split the labels into tokens. Given two tokens t_1 and t_2, we compute the maximum of the values returned by the following five techniques. (1) We assign a score of 0.0, if $t_1 = t_2$. (2) If t_1 and t_2 appear in the same synset in WordNet [11], we assign a score of -0.01. (3) We compute the Levenshtein distance [7], multiply it with -1 and assign any score higher than -0.2 to detect spelling variants. (4) If t_1 or t_2 is a single letter token and t_1 starts with t_2 or vice versa, we assign a score of -0.3. (5) We check if t_1 and t_2 have been modified at least two times by the same modifier. If this is the case, we assign a (very low) score of -0.9.

We have used the RockIt [14] Markov Logic engine to solve the optimization problem. RockIt does not support all logical constructs of our formalization directly. Thus, we had to rewrite existential quantification in terms of a comprehensive grounded representation. We applied our approach to the OAEI conference track. The results are depicted in Table 2.

2014	Pre	F	Rec	**2013**	Pre	F	Rec	**2012**	Pre	F	Rec
*	.80	.68	.59	YAM++ [12]	.78	.71	.65	YAM++	.78	.71	.65
AML [4]	.80	.67	.58	*	.80	.68	.59	*	.80	.68	.59
LogMap [6]	.76	.63	.54	AML	.82	.64	.53	LogMap	.77	.63	.53
XMAP [3]	.82	.57	.44	LogMap	.76	.63	.54	CODI	.74	.63	.55

Table 2. The proposed approach (*) compared with the top systems of 2012, 2013, and 2014.

We have listed the top-3 participants of the OAEI 2012, 2013, and 2014 conference track. The results are presented in term of precision (Pre), recall (Rec), and F-measure (F) using the the `ra2` reference alignment.[2] For each year the results are ordered by the F-measure that has been achieved. We inserted the results of our system, marked as *, at the appropriate row. Note that the vast majority of participating systems, which perform worse, is not depicted in the table. It can be seen that our approach is on the first position in 2014 and on the second in 2013 and 2012. This is a very good result, because we spent only a limited amount of work in the computation of the ingoing similarity scores. On the contrary, we presented above a complete description in less then 10 lines. This indicates that the quality of the generated alignments is mainly based our new approach for modeling the task of selecting the final alignment from the given similarity scores.

The OAEI conference dataset can processed in less than 20 minutes on a standard laptop. While slightly larger matching tasks are still feasible, significantly larger tasks cannot be solved anymore. Scalability is indeed an open challenge for the proposed approach. Currently we are working on a robust version of our approach in order to participate in the OAEI 2015 campaign.[3]

6 Conclusion

We presented a new approach for extracting a final alignment from an initial set of matching hypotheses. We have argued by a detailed discussion of several examples that our approach makes reasonable choices in situations where classical approaches are doomed to fail. Moreover, our approach generates results in a transparent and comprehensible manner. It can, for example, be proven that any other solution with a better objective must be invalid. Moreover, the objective for any other possible solution can be

[2] The `ra2` reference alignment is not available for the public. We thank Ondřej Šváb-Zamazal, one of the track organizers, for conducting an evaluation run outside an OAEI campaign.

[3] A first implementation is available at http://web.informatik.uni-mannheim.de/mamba/

computed to understand why the generated alignment was preferred over an alternative. A preliminary evaluation has shown that our approach can compete with the top systems participating in previous OAEI campaigns even though we put only limited effort in the optimal choice and design of the similarity measures we used in our evaluation. While the evaluation revealed that scalability is a crucial issue for the proposed approach, the positive results observed so far as well as the elegant nature of the approach engages us to improve the approach and to analyze it future work.

References

1. José-Luis Aguirre, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Ondrej Sváb-Zamazal, Cássia Trojahn dos Santos, Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Benjamin Zapilko. Results of the ontology alignment evaluation initiative 2012. In *Proceedings of the 7th International Workshop on Ontology Matching*, 2012.
2. Sivan Albagli, Rachel Ben-Eliyahu-Zohary, and Solomon E. Shimony. Markov network based ontology matching. *Journal of Computer and System Sciences*, 78(1):105–118, 2012.
3. Warith Eddine Djeddi and Mohamed Tarek Khadir. XMap++: Results for oaei 2014. In *Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference*, pages 163–169.
4. Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel Cruz, and Francisco Couto. The agreementmakerlight ontology matching system. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 527–541. Springer, 2013.
5. Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251, 2009.
6. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *The Semantic Web–ISWC 2011*, pages 273–288. Springer, 2011.
7. Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
8. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, University Mannheim, 2011.
9. Christian Meilicke, Jan Noessner, and Heiner Stuckenschmidt. Towards joint inference for complex ontology matching. In *AAAI (Late-Breaking Developments)*, 2013.
10. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.
11. George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
12. DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov. Extended tversky similarity for resolving terminological heterogeneities across ontologies. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 711–718. Springer, 2013.
13. Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *AAAI*, 2010.
14. Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. RockIt: Exploiting parallelism and symmetry for map inference in statistical relational models. 2013.
15. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.