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.

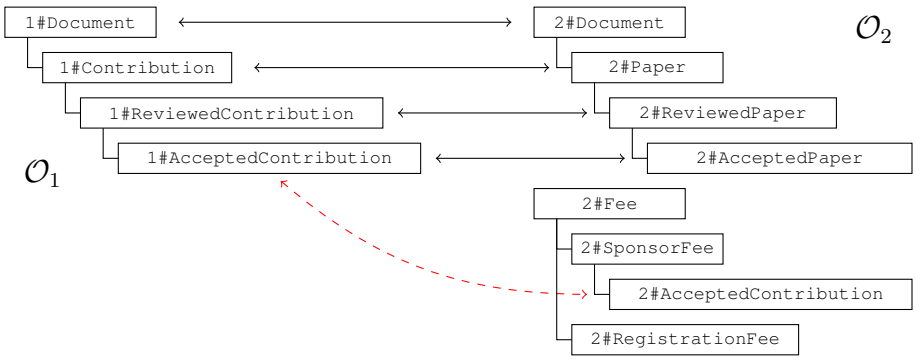


Fig. 1. Example of a non-trivial matching problem.

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.

A Multilingual Ontology Matcher

Gábor Bella*, Fausto Giunchiglia†, Ahmed AbuRa'ed†, and Fiona McNeill*

*Heriot-Watt University, †University of Trento

Abstract State-of-the-art multilingual ontology matchers use machine translation to reduce the problem to the monolingual case. We investigate an alternative, self-contained solution based on *semantic matching* where labels are parsed by multilingual natural language processing and then matched using a language-independent knowledge base acting as an interlingua. As the method relies on the availability of domain vocabularies in the languages supported, matching and vocabulary enrichment become joint, mutually reinforcing tasks. In particular, we propose a vocabulary enrichment method that uses the matcher's output to detect and generate missing items semi-automatically. Vocabularies developed in this manner can then be reused for other domain-specific natural language understanding tasks.

1 Introduction

Classification hierarchies, tree-structured data schemas, taxonomies, and term bases are widely used around the world as simple, well-understood, semi-formal data and knowledge organisation tools. They often play a normative role both as a means for classification (of documents, open data, books, items of commerce, web pages, etc.) and as sources of shared vocabularies for actors cooperating in a given domain. Activities such as international trade and mobility rely on the interoperability and integration of such resources across languages. Cross-lingual[1] ontology matching attempts to provide a solution for creating and maintaining alignments for such use cases.

State-of-the-art matchers that evaluate as the best in the *Multifarm* cross-lingual matching tasks of OAEI [6], such as AML [1] or LogMap [9], use online translation services (typically from Microsoft or Google) in order to reduce the problem of language diversity to the well-researched problem of monolingual English-to-English matching. The success of these methods is dependent on the availability of the translation service that is being used as a black box. Still, with the constant improvement of such services, matchers using machine translation are able to provide usable results and are able to deal with a wide range of languages.

In this paper we investigate a different perspective on cross-lingual matching that considers the building and maintenance of multilingual vocabularies as part

[1] We use the term *cross-lingual matching* as a specific case of multilingual matching when ontologies in two different languages are being aligned.

of the alignment task. The method is based on the use of locally available multilingual lexical-semantic *vocabularies*. Such resources are in constant evolution and are often available on the web with a more or less wide coverage of different terminological domains.

We are motivated by three considerations: first, we set out to explore to what extent such a linguistically-oriented, non-statistical approach to cross-lingual matching can be used as a viable alternative to machine translation. Secondly, we wish to provide a natively multilingual matcher that is entirely under the control of its user and does not rely on a non-free external translator service. This is necessary for high-value applications, such as e-commerce or libraries, where quality has to remain fully under the user's control. Finally, besides using vocabularies as resources for matching, we show how the matcher's output itself can become a resource in the purpose of vocabulary enrichment. This positive feedback loop exploits mismatches for increased terminological coverage which, in turn, improves subsequent matching results. One example use case is integration of open data—available in multiple languages—for mobility applications where geographical concepts and names are matched with the *GeoWordNet* catalogue [2].

While there is existing work [7] on using post-processing to repair a matching through the enrichment of background knowledge, our goal is different: we attempt to collect missing *vocabulary elements* that can be stored and subsequently reapplied, whereas [7] finds unknown *relations* between labels that may not be reusable outside the context of the matching task.

We took as basis for our work the SMATCH semantic matcher tool, for two main reasons: first, it operates on the level of meanings of labels instead of surface techniques, which makes it a suitable tool for cross-lingual semantic comparisons. Secondly, SMATCH is designed for matching *lightweight ontologies*, semi-formal knowledge organisation structures typically used for purposes of classification, that we believe are the main focus of most real-world cross-lingual matching challenges. Lightweight ontologies, as defined in [3], are characterised by (1) having a tree structure, (2) having nodes expressed as well-formed natural language labels, (3) they assume classification semantics (the extension of a node *Italy* under a node *Literature* are documents on Italian literature), and (4) the meaning of edges is not formally defined (they may stand for *is-a*, *part-of*, etc.).

The result of this work is NuSMATCH (NuSM for short), a first step in the direction of a new-generation multilingual matcher that has built-in capabilities for cross-lingual matching and that can also be used as a multilingual vocabulary enrichment tool.

The rest of the paper is organised as follows. Section 2 presents the *multilingual knowledge base*, the core resource for our matcher. Section 3 provides a brief reminder on semantic matching and on NuSM, while section 4 details our multilingual extensions. Section 5 presents vocabulary enrichment using erroneous mappings output by the matcher. Section 6 provides evaluation results and discussion, while section 7 presents issues not yet resolved.

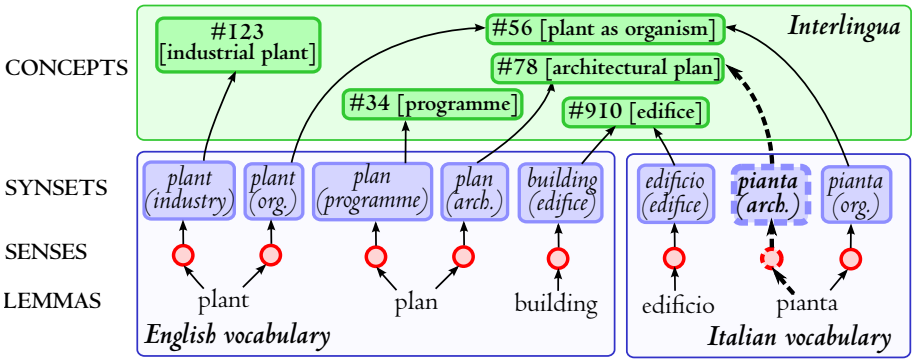


Figure 1. English and Italian vocabularies with the interlingua acting as a language-independent interoperability layer. The vocabularies may not be complete: the Italian sense and synset *pianta*, meaning 'architectural plan', is marked with dashed lines to indicate that it is missing from the Italian vocabulary.

2 A Multilingual Knowledge Base as Interlingua

Our approach to cross-lingual matching relies on a multilingual knowledge resource consisting of two layers: (1) a lower layer of multilingual *vocabularies* that are WordNet-like lexical-semantic resources; and (2) the *interlingua*: a language-independent ontology of concepts, each one linked to its corresponding vocabulary items in each language. This architecture has already been implemented at the University of Trento as part of a larger knowledge resource called the *Universal Knowledge Core* (UKC) [3], that we reuse for our purposes.

The architecture of a *vocabulary* is similar to that of Princeton WordNet [10], consisting of *lemmas* (i.e., dictionary forms of words of a language) associated to formally defined *word senses*. Synonymous senses are grouped together in synonym sets or *synsets*. Both senses and synsets are interconnected by lexical-semantic relations. Synsets represent an abstraction from the language-specific lexicon towards units of meaning and, indeed, the WordNet synset graph is sometimes used as an upper ontology for general reasoning tasks. This practice is suboptimal because of the known Anglo-Saxon cultural and linguistic bias of the synset graph (see, for example, [12]). As a solution, our multilingual knowledge base (simply *knowledge base* in the following) introduces the *interlingua* as a manually curated ontology representing a language-independent abstraction from the synset graph. Each synset in each vocabulary is mapped to a concept (fig. 1). The opposite is not necessarily true, e.g., when a vocabulary is incomplete. The interlingua acts as an interoperability layer across language-specific vocabularies, a feature that we use for cross-lingual matching.

High-quality vocabularies are costly to build in terms of human effort. Existing wordnets[2]—that we reuse to bootstrap our vocabularies when it is legally and technically possible—tend to be incomplete to a smaller or greater extent: for

[2] `http://globalwordnet.org/wordnets-in-the-world/`

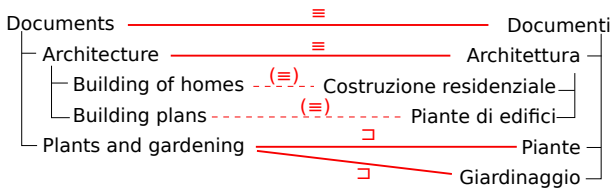


Figure 2. Example English and Italian classifications of documents, with some example mapping relations. Dashed lines with '(≡)' denote false negatives (mappings not found by the matcher), for reasons explained in section 5.

example, the Spanish *Multilingual Central Repository 3.0*[3] contains 56K lemmas and 38K synsets, the *Italian MultiWordNet*[4] contains 42K lemmas and 33K synsets, while Princeton WordNet 3.0 contains about 200K and 118K, respectively. Furthermore, wordnets tend to be general-purpose vocabularies that lack domain-specific terminology.

Efforts parallel to ours for building multilingual knowledge resources do exist. In earlier efforts such as EuroWordNet [11] or MCR [4] cross-lingual interoperability was provided by mapping non-English synsets to their English Princeton WordNet counterparts. This meant inheriting the English-centric lexical-semantic bias both in vocabulary construction and in reasoning. *BabelNet* [5] is a more recent and more advanced effort, with the same architectural design and underlying ideas as our knowledge base. The difference lies in the methodology of building it: BabelNet is mostly built automatically from diverse sources such as *Wikipedia* and *OmegaWiki*, while our knowledge base is built and maintained by human effort using both expert input and crowdsourcing. While the general problem of constructing lexical-semantic resources is beyond the scope of this paper, one of the outcomes of our work is a method for vocabulary enrichment using the output of NuSM.

3 NuSM

NuSM is designed as a multilingual extension of the SMATCH (English-only) semantic matcher [8]. Matching is semantic because, first, it is based on word senses extracted from ontology labels, secondly, it is performed using propositional logical inference and, thirdly, the mappings returned are description logic relations of equivalence, subsumption, and disjointness (for an example see fig. 2). We follow the basic four-step design of SMATCH, shown as pseudocode in fig. 3. Two new pre- and post-processing steps were added for language detection and for the semi-automated enrichment of vocabularies, respectively.

Below we provide a brief overview of each step of the matching process, followed by an in-depth discussion on the steps that are new or were modified.

[3] `http://adimen.si.ehu.es/web/MCR`
[4] `http://multiwordnet.fbk.eu`

	SMATCH	NuSM
step 0		`srcLang := detectLanguage(srcTree)` `trgLang := detectLanguage(trgTree)`
step 1	`computeLabelFormulas(srcTree)` `computeLabelFormulas(trgTree)`	`computeLabelFormulas(srcLang, srcTree)` `computeLabelFormulas(trgLang, trgTree)`
step 2	`computeNodeFormulas(srcTree)` `computeNodeFormulas(trgTree)`	

step 3	`for each srcAtom in srcTree:`
	`  for each trgAtom in trgTree:`
	`  wordNetMatcher(srcAtom, trgAtom)`
	`  stringMatcher(srcAtom, trgAtom)`
		`for each srcAtom in srcTree:`
		`  for each trgAtom in trgTree:`
		`  conceptMatcher(srcAtom, trgAtom)`
		`  nameMatcher(srcAtom, trgAtom)`

step 4	`mappings := treeMatcher(srcTree, trgTree)`	
step 5		`enrichVocabularies(mappings)`

Figure 3. Comparison of the high-level steps in SMATCH and NuSM.

For a more detailed presentation of semantic matching and the original SMATCH tool, we refer the reader to [8].

Step 0 is a new pre-processing step that detects the language of the two trees in input. We do not handle the rare case of ontologies mixing labels in multiple languages, as this would reduce the overall accuracy of language detection. Processing is interrupted if for the detected language no suitable vocabulary or NLP parser is available.

Step 1 computes *label formulas* for the two trees, that is, a propositional description logic formula corresponding to the semantic representation of the label. Atoms of the formula are sets of concepts from the interlingua, possibly representing the meaning of the atom, while operators are conjunctions, subjunctions, and negations. For example, in fig. 2, for the English label *Plants and gardening* the formula $plant \sqcup gardening$ is computed where *plant* and *gardening* are sets of concepts and the coordinating conjunction *and* becomes a disjunction (since the node classifies documents about any of the two topics). As for the label *Building plans*, it becomes a conjunctive formula: $building \sqcap plan$. The difference with respect to SMATCH is that label formulas are computed in a language-dependent manner, while meanings associated to the atoms are language-independent concepts from the interlingua instead of WordNet synsets.

Step 2 computes for each node tree their *node formulas*, which are formulas describing labels in the context of their ancestors. This step consists of computing for each label formula its conjunction with the label formulas of all of its ancestors. For *Plants and gardening*, this becomes $(plant \sqcup gardening) \sqcap document$. This step was not modified with respect to the original SMATCH.

Step 3 collects axioms relevant to the matching task. For each meaning in each atom of the source tree, step 3 retrieves all relations that hold between it and all meanings of all atoms in the target tree. In SMATCH, WordNet is used as a knowledge base (`wordNetMatcher` method) and additional axioms are inferred through string matching techniques (`stringMatcher` method). In NuSM, the interlingua is used as background knowledge (`conceptMatcher`) and string

matching is used mainly for names (`nameMatcher`). For example, for the pair of atoms (*plant*, *pianta*) retrieved from the interlingua in fig. 1, if both have a concept set of two concepts, this means retrieving potential relations for four concept pairs.

Step 4 performs the matching task (`treeMatcher` method) by running a SAT solver on pairs of source-target node formulas (f_S, f_T), computed in step 2 and complemented by corresponding axioms retrieved in step 3. If a pair turns out to be related by one of three relations: *equivalence* $f_S \leftrightarrow f_T$, *implication* $f_S \leftarrow f_T$ or $f_S \rightarrow f_T$, or *negated conjunction* $\neg(f_S \wedge f_T)$ then the mapping relation equivalence, subsumption, or disjointness is returned as a result, respectively. If none of the above holds, a no-match (*overlap*) relation is returned. This step was not modified with respect to the original SMATCH.

Step 5 is introduced specifically for NuSM as a post-processing step. Its goal is to discover mismatches resulting from missing vocabulary items, and help extend the vocabulary accordingly. For example, in fig. 2, no relation is returned between *Building plans* and *Piante di edifici* if the meaning 'plan' for *pianta* is missing from the Italian vocabulary.

4 Cross-Lingual Matching

In this section we explain how steps 1 and 3 were extended to adapt to cross-lingual operation.

4.1 Computing Label Formulas

The `computeLabelFormulas` method consists of three substeps: (1) building the label formula by parsing each label using language-specific NLP techniques; (2) computing of concept sets for each atom of the label formula; and (3) context-based sense filtering for polysemy reduction.

In NuSM, word senses in label formulas are represented by language-independent concepts from the interlingua. In order to compute label formulas and the concept sets of its atoms, language-dependent parsing is performed on labels.

Substep 1.1: label formulas are built by recognising words and expressions that are to be represented as atoms, and by parsing the syntactic structure of the label. For this purpose we use NLP techniques adapted to the specific task of ontology label parsing, distinguished by the shortness of text (typically 1-10 words) and a syntax that is at the same time limited (mostly noun, adjective, and prepositional phrases) and non-standard (varying uses of punctuation and word order). Depending on the language, different NLP techniques are used:

- word boundaries are identified through language-dependent tokenisation, e.g., *dell'/acqua* in Italian vs. *water/'s* in English, the apostrophe falling on different sides;
- language-dependent part-of-speech tagging helps in distinguishing open- and closed-class words where the former (nouns, verbs, adjectives, adverbs) become atoms while the latter (coordinating conjunctions, prepositions, punctuation, etc.) become logical operators;

English	Italian	Operator
except, non, without, ...	eccetto, escluso, non, senza, ...	$\neg$
and, or, ',', ...	e, o, ',', ...	$\sqcup$
of, to, from, against, for, ...	di, del, della, dello, dell', a, al, alla, allo, all', per, contro, ...	$\sqcap$

Figure 4. Mapping of closed-class words in labels to description logic operators (the list is incomplete).

- lemmatisation (morphological analysis of word forms in order to obtain the corresponding lemmas) is also performed using language-dependent methods, e.g., rule-based, dictionary-based, or the combination of the two;
- multiwords (e.g., *hot dog*) are recognised using dictionary lookup in the appropriate knowledge base vocabulary;
- closed-class words (pronouns, prepositions, conjunctions, etc.) and certain punctuation are mapped to the logical operators of conjunction, disjunction, and negation where mappings are defined for each language (cf. fig. 4);
- syntactic parsing—that determines how logical formulas are bracketed—is also done in a language-dependent manner.

Substep 1.2: concept sets are computed for each atom by retrieving from the interlingua all possible language-independent concepts for each open-class word appearing in the label. Thus, for the word *plant* we retrieve both the concept *plant as organism* and the concept *industrial plant* (fig. 1). What is new with respect to SMATCH is the language-independence of concepts and that concepts of derivationally related words are also retrieved, e.g., *plantation*, *planting*. This provides us increased robustness with respect to approximate grammatical correspondences between labels, a phenomenon that we observed as much more common in the cross-lingual than in the monolingual case (e.g., *piante di banane* vs. *banana plantation*).

Substep 1.3: sense filtering. In SMATCH, two atoms are by default considered equal if they have the same word form or lemma, regardless of the actual meanings: if the word *plant* appears both in the source and the target tree, they may be matched regardless of their respective meanings (*living organism* or *industrial building*). In order to reduce false positives due to such cases of polysemy, SMATCH implements a form of word sense disambiguation called *sense filtering*. This operation has a lesser importance in a cross-lingual scenario as the coincidence of homographs across languages is much rarer. For example, matching the English word *plant* with the Italian word *pianta*, both polysemous as shown in fig. 1, does not pose a problem as *pianta* does not have a meaning of 'industrial plant', nor does *plant* mean 'architectural plan'. This phenomenon acts as a 'natural' word sense disambiguation technique, allowing us to finetune recall by switching off the sense filtering algorithm implemented in SMATCH when the source and target languages are different and only apply it if the two languages are the same.

4.2 Retrieval of Axioms

SMATCH performs semantic matching between atoms by retrieving axioms as WordNet relations between senses and synsets (the `wordNetMatcher` method in fig. 3). NuSM, in contrast, relies on language-independent ontological relations existing in the interlingua (`conceptMatcher`). Equivalence is implied by concept equality and subsumption is derived from *is-a*, *attribute-value*, and *part-whole* relations, taking transitivity into account.

String similarity is a common metric used in monolingual matchers. SMATCH relies on string similarity between words and between glosses of WordNet synsets (the `stringMatcher` method includes both techniques) whenever WordNet does not provide any semantic axioms. Even though string similarity has a more limited scope of use in cross-lingual matching—words unrecognised because missing from the vocabularies cannot be assumed to match across different languages—we still use it for the matching of names and acronyms which tend to have a higher resemblance across languages (`nameMatcher`). We discarded gloss-based matching as these are not available for all vocabularies and the gloss-based matcher does not work on glosses written in different languages.

5 Vocabulary Enrichment

Term lists, taxonomies, and classifications, when available in multiple languages, are useful resources for the extraction of domain-specific terminology. The idea is to exploit incorrect mappings in order to identify the vocabulary elements missing for a given language and, consequently, to enrich them in a semi-automated manner, supervised by a human user.

Generally, we consider that mappings perceived by the user as incorrect can be explained by thee main phenomena: (1) the incompleteness of the knowledge base, (2) the design and limitations of the matcher (e.g., NLP errors or the inability to match rough translations such as *Building of homes* vs. *Costruzione residenziale*, 'residential construction') , and (3) modelling errors in the classifications themselves (example: *Gardening and landscaping* classified under *Gardening* results in two being inferred to be equivalent due to classification semantics).

In the following we concentrate on errors of type 1 and especially on missing vocabulary items: word forms, lemmas, senses, and synsets. We leave the problem of enrichment of the interlingua by concepts and relations for future work. We provide a semi-automated method that identifies errors stemming from an incomplete vocabulary and proposes a corresponding repair-by-enrichment action to the user. The semi-automated approach strikes a balance between reducing human effort and maintaining the high quality of vocabularies. It requires the contribution of a skilled person, ideally a data scientist, with a good knowledge of both languages.

Step 1: selection of the tree to process. In order to detect whether vocabulary enrichment is necessary, we either rely on a decision by the user or on a heuristic based on the number of unrecognised words found in one of the trees

being over a certain threshold. The goal is to select the tree that corresponds to the vocabulary poorer in terminological coverage: in the following we will call this tree the 'poor tree' and the other one the 'rich tree'. The repair process traverses the poor tree in depth-first order from the root, as the repair of a node affects all of its descendants.

Step 2: node-by-node identification of false negative mappings. False negatives, by definition, are true mappings not found by the matcher. Our repair method, however, relies on this information to identify missing vocabulary items. For this reason, we need to have access to ground truth in the form of equivalences and subsumptions. We propose three possible methods for obtaining ground truth:

- *user-provided*, e.g., by manually pointing out false negatives node by node during the traversal process.
- *Pre-existing:* a great number of lightweight ontologies are available on the web in multiple languages, often as industry standards of economic areas englobing multiple countries (in section 6 we provide concrete examples). These multilingual classifications can be seen as *fully aligned parallel corpora* and be used for vocabulary enrichment where the alignment provides ground truth.
- *Automatically obtained:* the (monolingual) SMATCH is run in parallel using a machine translation service as preprocessor. We automate the identification of false negatives by comparing the mappings output by both SMATCH and NuSM. Negatives output by NuSM that are positives for SMATCH are likely candidates for false negatives. We assume that precision is high (false positives are few) in the monolingual case—which is generally true, cf. the evaluations in [8]—and that the overlap of the positives of SMATCH and NuSM is not total, in other words, that the former is able to provide new positives to the latter. Our experiments showed this to be the case (cf. section 6).

Step 3: identification of the missing vocabulary item and repair. As an example for the repair process, let us take the labels *Building plans* and *Piante di edifici* from fig. 2. They are represented here as atoms containing their meanings retrieved from the interlingua in fig. 1:

Building #910 'edifice'	⊓	*plans* #78 'architectural plan' #34 'programme'	≡	*Piante* #56 'plant as organism'	⊓	*edifici* #910 'edifice'

Because of the missing sense and synset 'architectural plan' for the lemma *pianta*, indicated by dashed lines in fig. 1, the equivalence is missed by the matcher. In the repair scenario, however, we are supposing it to be provided as ground truth. Once such an erroneous mapping has been identified, repair proceeds through the substeps below.

Substep 3.1: pre-selection of atoms that are likely subjects for repair. For each false negative mapping identified while traversing the poor tree, the atoms of the corresponding label are analysed. Atoms of unrecognised words (word forms or lemmas) are given priority, as an unrecognised word is a trivial cause

of false negatives. In the absence of unrecognised words, all atoms of the label are selected. In our example, the word *piante* is a recognised word (it does have one meaning, 'plant as organism', in the vocabulary), thus both $atom_{piante}$ and $atom_{edifici}$ are pre-selected.

Substep 3.2: selection of repair candidates. A repair candidate is a pair (*preselected atom*, *repair concept*) that, when the repair concept is substituted into the atom, repairs the mapping so that the mapping relation corresponds to the ground truth. In our example, ($atom_{piante}$, 'architectural plan') is such a repair candidate. In substep 2 a small subset of *repair concepts* is selected, depending on the ground truth relation to be obtained. If the relation is equivalence then the set of repair concepts corresponds to the concepts appearing in the 'rich' node formula of the mapping. If the relation is more general (resp. less general) then it corresponds to the concepts appearing in the 'rich' node formula plus all of their ancestors (resp. descendants). The suitable (*atom*, *repair concept*) pairs are retained as *repair candidates*. For the node *Piante di edifici* two repair candidates are found: ($atom_{piante}$, 'programme') and ($atom_{piante}$, 'architectural plan'). No other substitution of any concept from the left-hand side into any atom on the right-hand side leads to equivalence.

Substep 3.3: identification of the missing vocabulary item and its creation. The user filters appropriate repair candidates by answering questions such as *'is meaning "architectural plan" suitable for word* piante *in this label?'*. Upon an affirmative answer, we find the missing vocabulary item(s) within the path between the repair concept and the surface word form of the atom. Repair ends by inserting newly created item(s) into the vocabulary (again upon user acceptance). In our case, the presence of an Italian synset connected to the concept of 'architectural plan' is verified. As it is missing, a new synset is created, together with a sense and links connecting the synset with the lemma *pianta*. The created items are the ones shown in dashed lines in fig. 1.

6 Evaluation and Discussion

Our evaluations were performed on two language pairs: English-Spanish and English-Italian. We used a diverse set of industrial and public multilingual classifications and term bases.[5] As these classifications are fully aligned across languages, they provide ground truth for equivalent mappings. However, because of the nature of semantic matching, other valid equivalences and subsumptions may be returned between non-aligned nodes. For example, *Forestry/Logging* and *Forestry/Logging/Logging* are equivalent nodes according to classification semantics (both are formalised as *forestry* $\sqcap$ *logging*), yet such relations are missing from our ground truth. Manual production of ground truth being beyond our means for the 2,600 nodes evaluted, we have simplified our evaluations in order to allow the automation of tests:

[5] NACE: Statistical Classification of Economic Activities in the European Community, Rev. 2 (`ec.europa.eu/eurostat/ramon/`), EUROVOC: the EU's multilingual thesaurus (`eurovoc.europa.eu`), UDC: Universal Decimal Classification (`udcc.org`).

Corpus	Lang.	# nodes per tree	Avg. label length	Avg. depth	NuSM Prec. ≡	NuSM Recall ≡	Google smatch Prec.	Google smatch Recall
EUROVOC	EN-ES	300	2.3	1	95.9%	47.0%	98.2%	73.5%
EUROVOC	EN-IT	300	2.2	1	97.7%	56.4%	97.9%	77.9%
NACE	EN-ES	880	5.9	3.5	75.9%	20.7%	82.0%	28.5%
NACE-ATECO	EN-IT	880	6.2	3.5	82.4%	20.1%	90.3%	21.7%
UDC	EN-ES	125	5.3	2.5	63.3%	24.8%	100%	19.2%
UDC	EN-IT	125	5.1	2.5	100%	20.8%	71.7%	26.4%

Figure 5. Cross-lingual evaluation results on parallel classifications. Also included are the scores obtained by the monolingual SMATCH coupled with Google Translate.

- only relations of equivalence, that is, only perfect matches are evaluated as positives (subsumptions and disjointness are discarded);
- all returned equivalences that are not in the ground truth and cannot be trivially mapped to it (by reordering labels or removing duplicate labels) are considered as false positives.

Our results are in fig. 5. We consider the scores as promising first results, especially given our conservative evaluation method. According to close scrutiny, mapping errors (false positives and negatives) were a consequence of the following factors:

- the Spanish and Italian vocabularies we used contain 32K and 42K words, respectively, unlike our 130K English vocabulary. Missing words, senses, and synsets reduce both recall and precision.
- a weak point of our current matcher is its multilingual syntactic parser, which often results in wrong bracketing in label formulas. The longer the labels the higher the probability of a parsing error, which explains the gradual performance degradation correlated with increased label lengths in our evaluation datasets.
- the most important cause of low recall figures is the high number of non-exact translations present in the data (similar to the example *Building of homes* vs. *Costruzione residenziale*) in fig. 2). Such linguistic 'fuzziness' is perhaps the hardest cross-lingual matching problem to tackle.

The last two columns in fig. 5 represent scores obtained by SMATCH when fed by Google-translated English text. These scores are somewhat higher, although by varying margins and not in all cases. This is explained by radically different underlying NLP techniques: machine translators are essentially statistical tools based on word n-grams and thus work well on rough translations where no word-by-word cross-lingual correspondence exists. On the other hand, the statistical nature of machine translation sometimes introduces translation errors. The hypothesis that the two different approaches yield partly different matching results is confirmed by preliminary quantitative evaluations that gave 38.7% (EUROVOC), 55.3% (NACE), and 45.8% (UDC) as the percentage of true positives that were *not* found by NuSM among those that *were* found by Google-SMATCH. This proves that the translation-based method for obtaining ground truth that we supposed in section 5 can effectively work.

7 Conclusions and Future Work

The results presented in this paper, both regarding cross-lingual matching and vocabulary enrichment, reflect work in progress, with improvements ongoing in several areas. Improved language-specific syntactic parsing of ontology labels is likely to have a big impact on our scores. In the repair method, we plan to extend the scope of repair to the interlingua, both to concepts and relations. Finally, given our results, we see a new line of research in combining the vocabulary-based technique presented here with machine translation. Our observation on the difference between the sets of true positives returned by the two techniques points in the direction of a potentially efficient ensemble method.

Acknowledgment We owe a big thanks to Aliaksandr Autayeu, one of the main developers and the current maintainer of monolingual SMATCH, for his advice and for his relentless work on keeping the tool up to date. We also acknowledge the *SmartSociety* project, funded by the 7th Framework Programme of the European Community.

References

1. Daniel Faria et al. The AgreementMakerLight Ontology Matching System. In Robert Meersman et al., editor, *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, volume 8185 of *Lecture Notes in Computer Science*, pages 527–541. Springer Berlin Heidelberg, 2013.
2. Fausto Giunchiglia et al. GeoWordNet: A Resource for Geo-spatial Applications. In *Proceedings of ESWC 2010*, pages 121–136.
3. Fausto Giunchiglia et al. Faceted Lightweight Ontologies. In *Conceptual Modeling: Foundations and Applications*, volume 5600. Springer Berlin Heidelberg, 2009.
4. J. Atserias et al. The MEANING Multilingual Central Repository. In *In Proceedings of the Second International WordNet Conference*, pages 80–210, 2004.
5. Maud Ehrmann et al. Representing Multilingual Data as Linked Data: the Case of BabelNet 2.0. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014*.
6. Zlatan Dragisic et al. Results of the Ontology Alignment Evaluation Initiative 2014. In *ISWC 2014, Riva del Garda, Trentino, Italy.*, pages 61–104, 2014.
7. Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Discovering Missing Background Knowledge in Ontology Matching. In *Proceedings of ECAI 2006, Riva Del Garda, Italy.*
8. Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko. Semantic Matching: Algorithms and Implementation. *J. Data Semantics*, 9:1–38, 2007.
9. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-Based and Scalable Ontology Matching. In *The Semantic Web – ISWC 2011*, volume 7031, pages 273–288. 2011.
10. George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November 1995.
11. Piek Vossen, editor. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
12. Piek Vossen, Wim Peters, and Julio Gonzalo. Towards a Universal Index of Meaning. In *SIGLEX99: Standardizing Lexical Resources*, pages 81–90, 1999.

Understanding a Large Corpus of Web Tables Through Matching with Knowledge Bases – An Empirical Study

Oktie Hassanzadeh, Michael J. Ward, Mariano Rodriguez-Muro, and Kavitha Srinivas

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{hassanzadeh,MichaelJWard,mrodrig,ksrinivs}@us.ibm.com

Abstract. Extracting and analyzing the vast amount of structured tabular data available on the Web is a challenging task and has received a significant attention in the past few years. In this paper, we present the results of our analysis of the contents of a large corpus of over 90 million Web Tables through matching table contents with instances from a public cross-domain ontology such as DBpedia. The goal of this study is twofold. First, we examine how a large-scale matching of all table contents with a knowledge base can help us gain a better understanding of the corpus beyond what we gain from simple statistical measures such as distribution of table sizes and values. Second, we show how the results of our analysis are affected by the choice of the ontology and knowledge base. The ontologies studied include DBpedia Ontology, Schema.org, YAGO, Wikidata, and Freebase. Our results can provide a guideline for practitioners relying on these knowledge bases for data analysis.

Keywords: Web Tables, Annotation, Instance-Based Matching

1 Introduction

The World Wide Web contains a large amount of structured data embedded in HTML pages. A study by Cafarella et al. [6] over Google's index of English documents found an estimated 154 million high-quality relational tables. Subsequent studies show the value of web tables in various applications, ranging from table search [15] and enhancing Web search [1, 3] to data discovery in spreadsheet software [2, 3] to mining table contents to enhance open-domain information extraction [7]. A major challenge in applications relying on Web Tables is lack of metadata along with missing or ambiguous column headers. Therefore, a content-based analysis needs to be performed to understand the contents of the tables and their relevance in a particular application.

Recently, a large corpus of web tables has been made publicly available as a part of the Web Data Commons project [12]. As a part of the project documentation [13, 14], detailed statistics about the corpus is provided, such as distribution

of the number of columns and rows, headers, label values, and data types. In this paper, our goal is to perform a semantic analysis of the contents of the tables, to find similarly detailed statistics about the kind of entity types found in this corpus. We follow previous work on recovering semantics of web tables [15] and column concept determination [8] and perform our analysis through matching table contents with instances of large cross-domain knowledge bases.

Shortly after we started our study, it became apparent that the results of our analysis do not only reflect the contents of tables, but also the contents and ontology structure of the knowledge base used. For example, using our approach in tagging columns with entity types (RDF classes) in knowledge bases (details in Section 2), we observe a very different distribution of tags in the output based on the knowledge base used. Figure 1 shows a "word cloud" visualization of the most frequent entity types using four different ontologies. Using only DBpedia ontology classes, the most dominant types of entities seem to be related to people, places, and organizations. Using only YAGO classes, the most frequent types are similar to those from DBpedia ontology results, but with more detailed breakdown and additional types such as "Event" and "Organism" that do not appear in DBpedia results. Freebase results on the other hand are very different, and clearly show a large number of music and media related contents in Web tables. The figure looks completely different for Wikidata results, showing "chemical_compound" as a very frequent type, which is not observed in Freebase or YAGO types. This shows the important role the choice of knowledge base and ontology plays in semantic data analysis.

In the following section, we briefly describe the matching framework used for the results of our analysis. We then revise some of the basic statistics provided by authors of the source data documentation [14], and then provide a detailed analysis of the entity types found in the corpus using our matching framework. We end the paper with a discussion on the results and a few interesting directions for future work.

2 Matching Framework

In this section, we briefly describe the framework used for matching table contents with instances in public cross-domain knowledge bases. Although implementation of this framework required a significant amount of engineering work to make it scale, the methods used at the core of the framework are not new and have been explored in the past. In particular, our MapReduce-based overlap analysis is similar to the work of Deng et al. [8], and based on an extension of our previous work on large-scale instance-based matching of ontologies [9]. Here, we only provide the big picture to help understanding the results of our analysis described in the following sections.

Figure 2 shows the overall matching framework. As input, we have the whole corpus of Web Tables as structured CSV files on one hand and a set of RDF knowledge bases which we refer to as *reference knowledge* on the other hand. Based on our previous work on data virtualization [10], we turn both

(a) DBpedia Ontology Tags

(b) DBpedia YAGO Classes Tags

(c) Freebase Type Tags

(d) Wikidata Type Tags

Fig. 1. Word Cloud of Most Frequent Column Tags

the tabular data and RDF reference knowledge into a common format and store them as key-values on HDFS. For tabular data, the key is a unique URI identifying a column in an input table, and the values are the values that appear in the column. For reference knowledge input, the key is the RDF class URI, and the values are the labels of instances of that class. For example, URI `rep://webtables/23793831_0_4377639018067805567.csv/company+name` represents column with header `company+name` in file `23793831_0_4377639018067805567.csv` in the input data. The values associated with this URI are contents of the column, which in this case is a list of company names. An example of reference knowledge URI is `http://dbpedia.org/ontology/Company` which is the DBpedia ontology class representing entities of type "Company". The values associated with this URI are labels of instances of this type, which means a list of all company names in DBpedia.

The similarity analysis component of the framework takes in the key-values and returns as output a table with each record associating a column in an input table with a tag which is an RDF class in reference knowledge, along with a confidence score. This tag indicates a similarity between values associated with the column and the class in input key-values, based on a similarity measure. Our system includes a large number of similarity functions but for the purpose of this study, we focus on one similarity measure that is very simple yet accurate and powerful for annotation of tables. Similar to Deng et al. [8], we refer to this

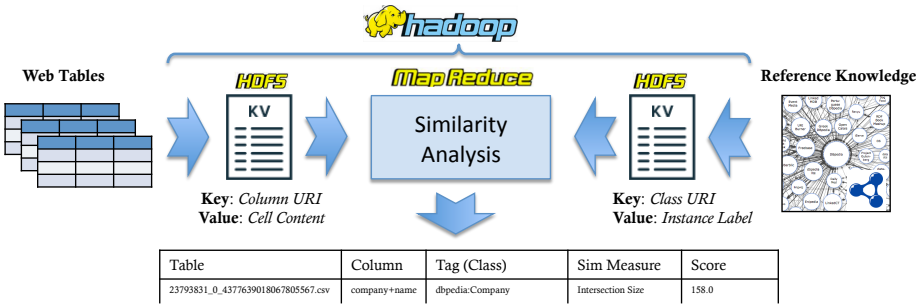


Fig. 2. Matching Framework

similarity analysis as *overlap analysis*. The values are first *normalized*, i.e., values are changed to lowercase and special characters are removed. We also filter numeric and date values to focus only on string-valued contents that are useful for semantic annotation. The similarity score is then the size of the intersection of the sets of filtered normalized values associated with the input URIs. The goal of overlap analysis is to find the number of values in a given column that represent a given entity type (class) in the input reference knowledge. In the above example, the column is tagged with class `http://dbpedia.org/ontology/Company` with score `158`, which indicates there are 158 values in the column that (after normalization) appear as labels of entities of type Company on DBpedia.

The *reference knowledge* in this study consists of three knowledge bases: (i) DBpedia [4] (ii) Freebase [5], and (iii) Wikidata [11, 16]. We have downloaded the latest versions of these sources (as of April 2015) as RDF NTriples dumps. DBpedia uses several vocabularies of entity types including DBpedia Ontology, Schema.org, and YAGO. We report the results of our analysis separately for these three type systems, which results in 5 different results for each analysis. We only process the English portion of the knowledge bases and drop non-English labels.

3 Basic Statistics

We first report some basic statistics from the Web Tables corpus we analyzed. Note that for this study, our input is the English subset of the Web Tables corpus [14] the same way we only keep the English portion of the reference knowledge. Some of the statistics we report can be found on the data publisher's documentation [14] as well, but there is a small difference between the numbers that could be due to different mechanisms used for processing the data. For example, we had to drop a number of files due to parsing errors or decompression failures, but that could be a results of the difference between the libraries used.

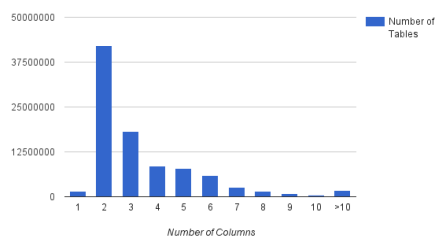
The number of tables we successfully processed is 91,357,232, that results in overall 320,327,999 columns (on average 3.5 columns per table). This results in 320,327,999 unique keys and 3,194,624,478 values (roughly 10 values per column) in the key-value input of Web Tables after filtering numerical and non-string

values for similarity analysis. DBpedia contains 369,153 classes, out of which 445 are from DBpedia Ontology, 43 are from Schema.org, and 368,447 are from YAGO. Freebase contains 15,576 classes, while Wikidata contains 10,250 classes. The number of values after filtering numeric and non-string values is 67,390,185 in DBpedia, 169,783,412 in Freebase, and Wikidata has 2,349,915 values. These numbers already show how different the knowledge bases are in terms of types and values.

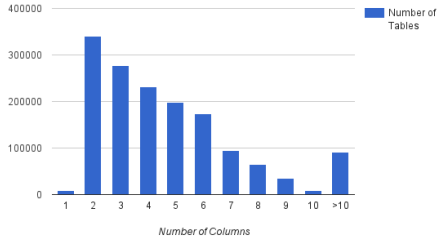
We first examine the distribution of rows and columns. Figure 3(a) shows the overall distribution of columns in the Web Tables. As it can be seen, the majority of the tables have lower than 3 columns. There are 1,574,872 tables with only 1 column, and roughly 62 million out of the 91 million tables (32%) have 2 or 3 columns. Now let us consider only the tables that appear in the output of our overlap analysis with intersection threshold set to 20, i.e., tables that in at least one of their columns have more than 20 normalized values shared with one of the knowledge reference sources. Such tables are much more likely to be of a higher quality and useful for further analysis and applications. Figure 3(b) shows the distribution of columns over these tables. As the figure shows, there is a smaller percentage of tables with small number of columns, with roughly 59% of the tables having 4 or more columns. This confirms the intuition that higher quality tables are more likely to have more number of columns, although there is still a significant number of tables with meaningful contents that have 3 or less columns.

Figure 3(c) shows the overall distribution of the number of rows in the whole corpus. Again, the majority of the tables are smaller ones, with roughly 78 million tables having under 20 rows, and roughly 1.5 million tables containing over 100 rows. Figure 3(d) shows the same statistics for tables with an overlap score over 20. Here again, the distribution of rows is clearly different from the whole corpus, with the majority of the tables having over 100 rows.

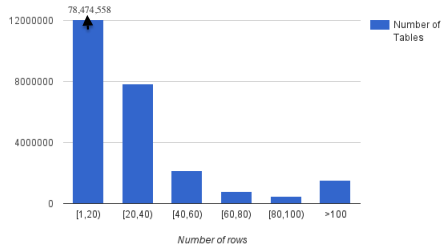
Next, we study the distribution of overlap scores over all tables and across different ontologies. Figure 4 shows the results (Schema.org results omitted for brevity). In all cases, the majority of tags have a score under 40, but there is a notable percentage of tags with a score above 100, i.e., the column has over 100 values shared with the set of labels of at least one type in the reference knowledge, a clear indication that the table is describing entities of that type. The main difference in the results across different ontologies is in the overall number of tags. With overlap score threshold of 20, there are 1,736,531 DBpedia Ontology tags, 542,178 Schema.org, 6,319,559 YAGO, 26,620,967 Freebase, and 865,718 Wikidata tags. The number of tags is a function of the size of the ontology in terms of number of classes and instances, but also the type system in the ontology. For example, Schema.org has only 43 classes resulting in an average of over 12,600 columns per each tag, but YAGO contains 368,447 classes which means an average of 17 columns per tag.



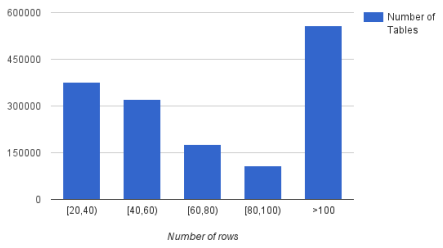
(a) Distribution of Number of Columns per Table



(b) Distribution of Number of Columns for Tables with 20+ Overlap Score



(c) Distribution of Number of Rows per Table



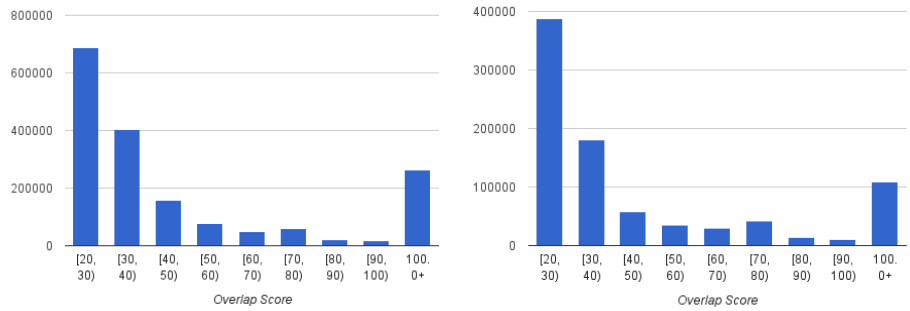
(d) Distribution of Number of Rows per Table with 20+ Overlap Score

Fig. 3. Distribution of Number of Rows and Columns

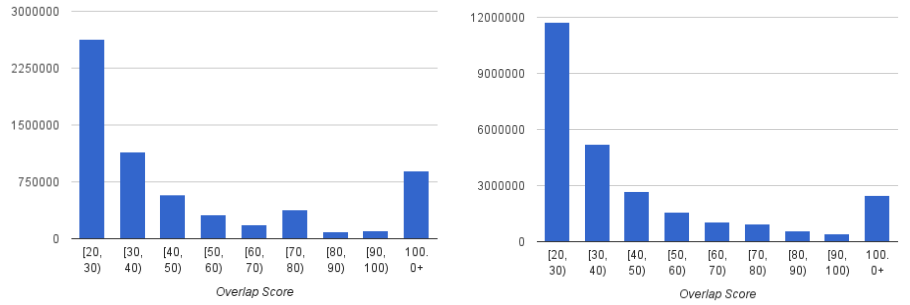
4 Distribution of Entity Types

We now present detailed statistics on the tags returned by the overlap similarity analysis described in Section 2. Going back to Figure 1 in Section 1, the word cloud figures are generated using the overlap analysis with the overlap threshold set to 20. The figure is then made using the top 150 most frequent tags in the output of the overlap analysis, with the size of each tag reflecting the number of columns annotated with that tag. The labels are derived either from the last portion of the class URI (for DBpedia and Freebase), or by looking up English class labels (for Wikidata). For example, "Person" in Figure 1(a) represents class `http://dbpedia.org/ontology/Person` whereas `music.recording` in Figure 1(c) represents `http://rdf.freebase.com/ns/music.recording`, and `chemical_compound` in Figure 1(d) represents `https://www.wikidata.org/wiki/Q11173` which has "chemical compound" as its English label.

In addition to the word cloud figures, Tables 1 and 2 show the top 20 most frequent tags in the output of our similarity analysis for each of the ontologies, along with their frequency in the output. From these results, it is clear that no single ontology on its own can provide the full picture of the types of entities that can be found on the Web tables. DBpedia ontology seem to have a better



(a) DBpedia Ontology (d) Wikidata



(a) YAGO (b) Freebase

Fig. 4. Distribution of Overlap Scores in Different Ontologies

coverage for person and place related entities, whereas YAGO has a large number of abstract classes being most frequent in the output. Schema.org provides a cleaner view over the small number of types it contains. Wikidata has a few surprising types on the top list, such as "commune of France". This may be due to a bias on the source on the number of editors contributing to entities under certain topics. Freebase clearly has a better coverage for media-related types, and the abundance of tags in music and media domain shows both the fact that there is a large number of tables in the Web tables corpus containing music and entertainment related contents, and that Freebase has a good coverage in this domain.

Finally, we examine a sample set of entity types across knowledge bases and see how many times they appear as a column tag in the overlap analysis output. Table 3 shows the results. Note that we have picked popular entity types that can easily be mapped manually. For example, Person entity type is represented by class `http://dbpedia.org/ontology/Person` in DBpedia, `http://dbpedia.org/class/yago/Person` in YAGO, `http://schema.org/Person` in Schema.org and

Table 1. Most Frequent Tags in DBpedia Ontology, YAGO, and Schema.org

DBpedia Ontology		YAGO		Schema.org	
Type	**Freq.**	**Type**	**Freq.**	**Type**	**Freq.**
Agent	242,410	PhysicalEntity	364,830	Person	186,332
Person	186,332	Object	349,139	Place	120,361
Place	120,361	YagoLegalActorGeo	344,487	CreativeWork	53,959
PopulatedPlace	112,647	Whole	230,667	Organization	50,509
Athlete	85,427	YagoLegalActor	226,633	Country	37,221
Settlement	60,219	YagoPerm.LocatedEntity	198,304	MusicGroup	22,926
ChemicalSubstance	57,519	CausalAgent	186,789	EducationalOrg.	12,159
ChemicalCompound	57,227	LivingThing	182,570	City	10,743
Work	53,959	Organism	182,569	CollegeOrUniversity	10,598
Organisation	50,509	Person	175,501	Movie	10,243
OfficeHolder	40,198	Abstraction	145,407	SportsTeam	9,594
Politician	39,121	LivingPeople	136,955	MusicAlbum	4,786
Country	37,221	YagoGeoEntity	120,433	Book	2,103
BaseballPlayer	30,301	Location	109,739	School	1,181
MotorsportRacer	26,293	Region	106,200	MusicRecording	1,166
RacingDriver	25,135	District	95,294	Product	1,130
Congressman	24,143	AdministrativeDistrict	92,808	TelevisionStation	1,037
MusicalWork	17,881	Group	85,668	StadiumOrArena	918
NascarDriver	16,766	Contestant	60,177	AdministrativeArea	896
Senator	15,087	Player	56,373	RadioStation	815

`http://rdf.freebase.com/ns/people.person` in Freebase. The numbers show a notable difference between the number of times these classes appear as column tags, showing a different coverage of instances across the knowledge bases. Freebase has by far the largest number of tags in these sample types. Even for the three ontologies that have the same instance data from DBpedia, there is a difference between the number of times they are used as a tag, showing that for example there are instances in DBpedia that have type Person in DBpedia ontology and Schema.org but not YAGO, and surprisingly, there are instances of Country class type in YAGO that are not marked as Country in DBpedia ontology or Schema.org.

5 Conclusion & Future Directions

In this paper, we presented the results of our study on understanding a large corpus of web tables through matching with public cross-domain knowledge bases. We focused on only one mechanism for understanding the corpus of tables, namely, tagging columns with entity types (classes) in knowledge bases. We believe that our study with its strict focus can provide new insights into the use of public cross-domain knowledge bases for similar analytics tasks. Our results clearly show the difference in size and coverage of domains in public cross-domain knowledge bases, and how they can affect the results of a large-scale analysis. Our results also show several issues in the Web Data Commons Web Tables corpus, such as the relatively large number of tables that contain very little or no meaningful contents.

Our immediate next step includes expanding this study to include other similarity measures and large-scale instance matching techniques [9]. Another interesting direction for future work is studying the use of domain-specific knowledge

Table 2. Most Frequent Tags in Wikidata and Freebase

Wikidata		Freebase	
Type	**Freq.**	**Type**	**Freq.**
Wikimedia_category	146,024	music.release_track	968,121
human	93,544	music.recording	964,906
chemical_compound	52,380	music.single	950,099
sovereign_state	34,681	location.location	532,053
country	22,030	people.person	475,472
determinator_for_. . . _occurrence	13,354	location.dated_location	460,766
city	12,823	location.statistical_region	458,643
commune_of_France	10,459	tv.tv_series_episode	440,985
taxon	10,127	location.citytown	409,315
landlocked_country	8,899	music.artist	390,458
island_nation	7,439	fictional_universe.fictional_character	372,820
republic	7,431	film.film_character	344,755
university	4,083	music.album	314,494
town	3,467	music.release	306,857
American_football_club	3,207	media_common.creative_work	304,231
band	3,024	media_common.cataloged_instance	297,875
municipality_of_Spain	2,950	type.content	269,216
comune_of_Italy	2,531	common.image	269,213
basketball_team	2,041	book.written_work	248,902
municipality_of_Germany	1,923	book.book	235,165

Table 3. Sample Entity Types and Their Frequency in Overlap Analysis Tags

Type	**DBpedia Ontology**	**YAGO**	**Schema.org**	**Wikidata**	**Freebase**
Person	186,332	175,501	186,332	93,544	475,472
Company	12,066	11,770	–	1,831	68,710
Location	120,361	109,739	120,36	–	532,053
Country	37,221	39,338	37,221	22,030	39,316
Film	10,243	9,080	10,243	348	175,460

bases to study the coverage of a certain domain in the corpus of Web Tables. For example, biomedical ontologies can be used in matching to discover healthcare related structured data on the Web.

The results reported in this paper may change after the reference knowledge sources or the corpus of tables are updated. Therefore, our plan is to maintain a website containing our latest results, along with the output of our analysis that can be used to build various search and discovery applications over the Web Tables corpus[1].

References

1. Google Web Tables. `http://research.google.com/tables`. [Online; accessed 29-04-2015].
2. Microsoft Excel Power Query. `http://office.microsoft.com/powerbi`. [Online; accessed 29-04-2015].
3. S. Balakrishnan, A. Y. Halevy, B. Harb, H. Lee, J. Madhavan, A. Rostamizadeh, W. Shen, K. Wilder, F. Wu, and C. Yu. Applying WebTables in Practice. In *CIDR*, 2015.

[1] For latest results, refer to our project page: `http://purl.org/net/webtables`.

4. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A Crystallization Point for the Web of Data. *JWS*, 7(3):154–165, 2009.
5. K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.
6. M. J. Cafarella, A. Y. Halevy, D. Zhe Wang, E. Wu, and Y. Zhang. WebTables: Exploring the Power of Tables on the Web. *PVLDB*, 1(1):538–549, 2008.
7. B. B. Dalvi, W. W. Cohen, and J. Callan. WebSets: extracting sets of entities from the web using unsupervised information extraction. In *WSDM*, pages 243–252, 2012.
8. D. Deng, Y. Jiang, G. Li, J. Li, and C. Yu. Scalable Column Concept Determination for Web Tables Using Large Knowledge Bases. *PVLDB*, 6(13):1606–1617, 2013.
9. S. Duan, A. Fokoue, O. Hassanzadeh, A. Kementsietsidis, K. Srinivas, and M. J. Ward. Instance-Based Matching of Large Ontologies Using Locality-Sensitive Hashing. In *ISWC*, pages 49–64, 2012.
10. J. B. Ellis, A. Fokoue, O. Hassanzadeh, A. Kementsietsidis, K. Srinivas, and M. J. Ward. Exploring Big Data with Helix: Finding Needles in a Big Haystack. *SIGMOD Record*, 43(4):43–54, 2014.
11. F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandecic. Introducing Wikidata to the Linked Data Web. In *ISWC*, pages 50–65, 2014.
12. H. Mühleisen and C. Bizer. Web Data Commons - Extracting Structured Data from Two Large Web Corpora. 2012.
13. P. Ristoski, O. Lehmberg, R. Meusel, C. Bizer, A. Diete, N. Heist, S. Krstanovic, and T. A. Knller. Web Data Commons - Web Tables. `http://webdatacommons.org/webtables`. [Online; accessed 29-04-2015].
14. P. Ristoski, O. Lehmberg, H. Paulheim, and C. Bizer. Web Data Commons - English Subset of the Web Tables Corpus. `http://webdatacommons.org/webtables/englishTables.html`. [Online; accessed 29-04-2015].
15. P. Venetis, A. Y. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering Semantics of Tables on the Web. *PVLDB*, 4(9):528–538, 2011.
16. D. Vrandecic and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

Combining Sum-Product Network and Noisy-Or Model for Ontology Matching

Weizhuo Li

Institute of Mathematics,Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, P. R. China
liweizhuo@amss.ac.cn

Abstract. Ontology matching is the key challenge to achieve semantic interoperability in building the Semantic Web. We present an alternative probabilistic scheme, called GMap, which combines the sum-product network and the noisy-or model. More precisely, we employ the sum-product network to encode the similarities based on individuals and disjointness axioms across ontologies and calculate the contributions by the maximum a posterior inference. The noisy-or model is used to encode the probabilistic matching rules, which are independent of each other as well as the value calculated by the sum-product network. Experiments show that GMap is competitive with many OAEI top-ranked systems. Futhermore, GMap, benefited from these two graphical models, can keep inference tractable in the whole matching process.

1 Introduction

Ontology matching is the process of finding relationships or correspondences between entities of different ontologies[5]. Many efforts have been conducted to automate the discovery in this process, e.g., incorporating more elaborate approaches including scaling strategies[3, 6], ontology repair techniques to ensure the alignment coherence[8], employing machine learning techniques[4], using external resources to increase the available knowledge for matching[2] and utilizing probabilistic graphical models to describe the related entities[1, 10, 11].

In this paper, we propose an alternative probabilistic schema, called GMap, based on two special graphical models—sum-product network(SPN) and noisy-or model. SPN is a directed acyclic graph with variables as leaves, sums and products as internal nodes, and weighted edges[12]. As it can keep inference tractable and describe the context-specific independence[12], we employ it to encode the similarities based on individuals and disjointness axioms and calculate the contributions by the maximum a posterior inference. Noisy-or model is a special kind of Bayesian Network[9]. When the factors are independent of each other, it is more suitable than other graphical models, specially in the inference efficiency[9]. Hence, we utilize it to encode the probabilistic matching rules. Thanks to the tractable inference of these special graphical models, GMap can keep inference tractable in the whole matching process. To evaluate GMap, we adopt the data sets from OAEI ontology matching campaign. Experimental results indicate that GMap is competitive with many OAEI top-ranked systems.

2 Methods

In this section, we briefly introduce our approach. Given two ontologies O_1 and O_2, we calculate the lexical similarity based on edit-distance, external lexicons and TFIDF[5]. Then, we employ SPN to encode the similarities based on individuals and disjointness axioms and calculate the contributions. After that, we utilize the noisy-or model to encode the probabilistic matching rules and the value calculated by SPN. With one-to-one constraint and crisscross strategy in the refine module, GMap obtains initial matches. The whole matching procedure is iterative. If it does not produce new matches, the matching is terminated.

2.1 Using SPN to encode individuals and disjointness axioms

In open world assumption, individuals or disjointness axioms are missing at times. Therefore, we define a special assignment—"$Unknown$" for the similarities based on these individuals and disjointness axioms.

For the similarity based on individuals, we employ the string equivalent to judge the equality of them. When we calculate the similarity of concepts based on individuals across ontologies, we regard individuals of each concept as a set and use Ochiai coefficient[1] to measure the value. We use a boundary t to divide the value into three assignments(i.e., 1, 0 and $Unknown$). Assignment 1(or 0) means that the pair matches(or mismatches). If the value ranges between 0 and t or the individuals of one concept are missing, the assignment is $Unknown$.

For the similarity based on disjointness axioms, we utilize these axioms and subsumption relations within ontologies and define some rules to determine its value. For example, x_1, y_1 and x_2 are concepts that come from O_1 and O_2. If x_1 matches x_2 and x_1 is disjoint with y_1, then y_1 is disjoint with x_2. The similarity also have three assignments. Assignment 1(or 0) means the pair mismatches(or overlaps). Otherwise, the similarity based on disjointness axioms is $Unknown$.

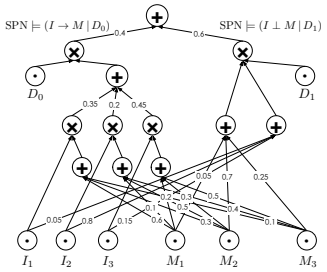


Fig. 1: The designed sum-product network

As shown in Figure 1, we designed a sum-product network S to encode above similarities and calculate the contributions, where M represents the contributions and leaves M_1, M_2, M_3 are indicators that comprise the assignments of M. All the indicators are binary-value. $M_1 = 1$(or $M_2 = 1$) means that the contributions are positive(or negative). If $M_3 = 1$, the contributions

[1] https://en.wikipedia.org/wiki/Cosine_similarity

are $Unknown$. Leaves I_1, I_2, I_3, D_0, D_1 are also binary-value indicators that correspond to the assignments of similarities based on individuals(I) and disjointness axioms(D). The concrete assignment metrics are listed in Table 1–2.

Table 1: Metric for Similarity D

Assignments	Indicators
$D = 1$	$D_0 = 0, D_1 = 1$
$D = 0$	$D_0 = 1, D_1 = 0$
$D = Unknown$	$D_0 = 1, D_1 = 1$

Table 2: Metric for Similarity I

Assignments	Indicators
$I = 1$	$I_1 = 1, I_2 = 0, I_3 = 0$
$I = 0$	$I_1 = 0, I_2 = 1, I_3 = 0$
$I = Unknown$	$I_1 = 0, I_2 = 0, I_3 = 1$

With the maximum a posterior(MAP) inference in SPN[12], we can obtain the contributions M. As the network S is complete and decomposable, the inference in S can be computed in time linear in the number of edges[7].

2.2 Using Noisy-Or model to encode probabilistic matching rules

We utilize probabilistic matching rules to describe the influences among the related pairs across ontologies and some of rules are listed in Table 3.

Table 3: The probabilistic matching rules among the related pairs

ID	Category	Probabilistic matching rules
R_1	class	two classes probably match if their fathers match
R_2	class	two classes probably match if their children match
R_3	class	two classes probably match if their siblings match
R_4	class	two classes about domain probably match if related objectproperties match and range of these property match
R_5	class	two classes about range probably match if related objectproperties match and domain of these properties match
R_6	class	two classes about domain probably match if related datapproperties match and value of these properties match

When we focus on calculating the matching probability of one pair, the matching rules are independent of each other as well as the value calculated by SPN. Therefore, we utilize the noisy-or model to encode them.

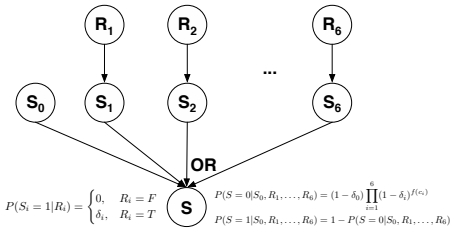


Fig. 2: The network structure of noisy-or model designed in GMap

Figure 2 shows the designed network, where R_i corresponds to the ith rule and S_i is the conditional probability depended on the condition of R_i. S_0 represents the SPN-based similarity that is a leak probability[9]. The matching

probability of one pair, $P(S = 1|S_0, R_1, ..., R_6)$, is calculated according to the formulas in the lower-right corner. c_i is the count of satisfied R_i and sigmoid function $f(c_i)$ is used to limit the upper bound of contribution of R_i. As the inference in the noisy-or model can be computed in time linear in size of nodes[9], GMap can keep inference tractable in the whole matching process.

3 Evaluation

To evaluate our approach[2], we adopt three tracks(i.e., Benchmark, Conference and Anatomy) from OAEI ontology matching campaign in 2014[3].

3.1 Comparing against the OAEI top-ranked systems

Table 4 shows a comparison of the matching quality of GMap and other OAEI top-ranked systems, which indicates that GMap is competitive with these promising existent systems. For Anatomy track, GMap does not concentrate on language techniques and it emphasizes one-to-one constraint. Both of them may cause a low alignment quality. In addition, all the top-ranked systems employ alignment debugging techniques, which is helpful to improve the quality of alignment. However, we do not employ these techniques in the current version.

Table 4: The comparison of GMap with the OAEI top-ranked systems

	Benchmark(Biblio)			Conference			Anatomy		
System	P	R	F	P	R	F	P	R	F
AML	**0.92**	0.4	0.55	**0.85**	0.64	**0.73**	**0.956**	**0.932**	**0.944**
LogMap	0.39	0.4	0.39	0.8	0.59	0.68	0.918	0.846	0.881
XMAP	1	0.4	0.57	0.87	0.49	0.63	0.94	0.85	0.893
CODI	n/a	n/a	n/a	0.74	0.57	0.64	0.967	0.827	0.891
GMap	0.63	**0.57**	**0.60**	0.67	**0.66**	0.66	0.930	0.802	0.862

3.2 Evaluating the contributions of these two graphical models

We separate SPN and the noisy-or model from GMap and evaluate their contributions respectively. As listed in Table 5, SPN is suitable to the matching task that the linguistic levels across ontologies are different and both of ontologies use same individuals to describe the concepts such as Biblio(201–210) in Benchmark track. Thanks to the contributions of individuals and disjointness axioms, SPN can improve the precision of GMap. When the structure information is very rich across the ontologies, the noisy-or model is able to discover some hidden matches with the existing matches and improve the recall such as in Anatomy track. However, if the ontology does not contain above features such as in Conference track, the improvement is not evident. Nevertheless, thanks to the complementary of these two graphical models to some extent, combining the sum-product network and the noisy-or model can improve the alignment quality as a whole.

[2] The software and results are available at https://github.com/liweizhuo001/GMap.
[3] http://oaei.ontologymatching.org/2014/

Table 5: The contributions of the sum-product network and the noisy-or model

	Biblio(201-210)			Conference			Anatomy		
System	P	R	F	P	R	F	P	R	F
string equivalent	0.680	0.402	0.505	0.8	0.43	0.56	**0.997**	0.622	0.766
lexical similarity(ls)	0.767	0.682	0.722	0.666	0.657	0.661	0.929	0.752	0.831
ls+spn	0.776	0.685	0.728	0.667	0.657	0.661	0.930	0.752	0.832
ls+noisy-or	0.782	0.701	0.739	**0.667**	**0.660**	**0.663**	0.937	0.772	0.847
ls+spn+noisy-or	**0.794**	**0.703**	**0.746**	**0.667**	**0.660**	**0.663**	0.930	**0.803**	**0.862**

4 Conclusion and Future Work

We have presented GMap, which is suitable for the matching task that many individuals and disjointness axioms are declared or the structure information is very rich. However, it still has a lot of room for improvement. For example, language techniques is essential to improve the quality of initial matches. In addition, dealing with alignment incoherent is also one of our future works.

Acknowledgments. This work was supported by the Natural Science Foundation of China (No. 61232015). Many thanks to Songmao Zhang, Qilin Sun and Yuanyuan Wang for their helpful discussion on the design and implementation of the GMap.

References

1. Albagli, S., Ben-Eliyahu-Zohary, R., Shimony, S.E.: Markov network based ontology matching. Journal of Computer and System Sciences 78(1), 105–118 (2012)
2. Zhang, S., Bodenreider, O.: Experience in aligning anatomical ontologies. International journal on Semantic Web and information systems 3(2), 1–26 (2007)
3. Djeddi, W.E, Khadir, M.T.: XMAP: a novel structural approach for alignment of OWL-full ontologies. In: Proc. of Machine and Web Intelligence(ICMWI). pp. 368–373 (2010)
4. Doan, A.H, Madhavan, J., Dhamankar, R., et al.: Learning to match ontologies on the semantic web. The VLDB Journal 12(4), 303–319 (2003)
5. Euzenat, J., Shvaiko, P.: Ontology Matching(2nd Edition). Springer (2013)
6. Faria, D., Pesquita, C., Santos, E., et al.: The agreementmakerlight ontology matching system In: 2013 OTM Conferences. pp. 527–541 (2013)
7. Gens, R., Pedro, D.: Learning the structure of sum-product networks. In: Proc. of International Conference on Machine Learning(ICML). pp. 873–880 (2013)
8. Jimenez-Ruiz, E., Grau, B.C.: LogMap: Logic-based and scalable ontology matching. In: Proc. of International Semantic Web Conference(ISWC). pp.273–288 (2011)
9. Koller, D., Friedman, N.: Probabilistic Graphical Models. MIT press (2009)
10. Mitra, P., Noy, N.F., Jaiswal, A.R. OMEN: A probabilistic ontology mapping tool. In: Proc. of International Semantic Web Conference(ISWC). pp. 537–547 (2005)
11. Niepert, M., Noessner, J., Meilicke, C., Stuckenschmidt, H.: Probabilistic-logical web data integration. Reasoning Web. pp. 504–533 (2011)
12. Poon, H., Domingos, P.: Sum-product networks: A new deep architecture. In: Proc of International Conference on Computer Vision Workshops(ICCV Workshops). pp. 689–690 (2011)

Towards Combining Ontology Matchers via Anomaly Detection

Alexander C. Müller and Heiko Paulheim

University of Mannheim, Germany
Research Group Data and Web Science
heiko@informatik.uni-mannheim.de,alexanda@mail.uni-mannheim.de

Abstract. In ontology alignment, there is no single best performing matching algorithm for every matching problem. Thus, most modern matching systems combine several *base matchers* and aggregate their results into a final alignment. This combination is often based on simple voting or averaging, or uses existing matching problems for learning a combination policy in a *supervised* setting. In this paper, we present the *COMMAND* matching system, an *unsupervised* method for combining base matchers, which uses anomaly detection to produce an alignment from the results delivered by several base matchers. The basic idea of our approach is that in a large set of potential mapping candidates, the scarce actual mappings should be visible as anomalies against the majority of non-mappings. The approach is evaluated on different OAEI datasets and shows a competitive performance with state-of-the-art systems.

Keywords: Ontology Alignment, Anomaly Detection, Outlier Detection, Matcher Aggregation, Matcher Selection

1 Introduction

In ontology matching, there is only rarely a *one size fits all* solution. Ontology matching problems differ along many dimensions, so that a matching system that performs well on one dataset does not necessarily deliver good results on another one. To overcome this problem, many ontology matching tools combine the results of various base matchers, i.e., individual matching strategies. However, this approach gives way to a new problem, i.e., how to *combine* the results of the base matchers in a way that the combination suits the problem at hand [7]. Solutions proposed in the past range from simple voting to supervised learning.

In this paper, we propose to use *anomaly* or *outlier detection* for the problem of matcher combination. Anomaly detection is the task of finding those data points in a data set that deviate from the majority of the data [1]. The underlying assumption is that given a large set of mapping candidates (e.g., the cross product of ontology elements from the ontologies at hand), the *actual* mappings (which are just a few) should stand out in one way or the other. Thus, it should be possible to discover them using anomaly detection methods. We show that it is possible to build a competitive matching system combining the results of more than 25 base matchers using anomaly detection.

2 Approach

COMMAND is a novel approach for dynamically selecting and combining ontology matchers via anomaly detection. The overall architecture is depicted in Fig. 1. The platform was implemented in Scala, the code is available on github under an open-source license.[1]

2.1 Base Matching and Matcher Selection

First, all base matchers that are based on local information of each ontology entity are executed. The entities of the target and source ontology are matched in a pair-wise fashion. This step matches *Classes*, *DataProperties* and *ObjectProperties* pairwise and independently.

After this the first *feature vector* is analyzed and an uncorrelated feature subset is extracted. The results of those uncorrelated matchers are used as the input similarities for the *structural matchers*.

The result of the *structural matchers* is joined with the element level matcher result to create a *feature vector*. Since some of the features might be redundant or not vary in their values and thus do not contribute to the final matching, we remove results with little variation, correlated results, and also support PCA for computing meaningful linear combinations of base matcher results.

The current version of COMMAND implements a large variety of element and structure level techniques. Those encompass 16 string similarity metrics, five external metrics based on WordNet and corpus linguistics, and five structural matching techniques, such as similarity flooding.

2.2 Aggregation by Anomaly Detection

The next step is the aggregation of the base matcher results into a final matching score for all correspondences. We perform this step by detecting outlying datapoints in the *feature vector space*, and using this score as a measure of similarity. The anomaly analysis and score normalization are performed separately for classes, data properties, and object properties.

To compute outlier scores, we apply *anomaly analysis techniques* on the feature vector representations. In this paper, we use three different techniques: A *k-nearest-neighbor based method (KNN)* that computes the anomaly score of a data point based on the average euclidean distances[2] to its nearest neighbors, a cluster-based method that calculates the unweighted *cluster-based local anomaly factor (CBLOF)* based on a given clustering scheme produced by an arbitrary clustering algorithm [5], and the *Replicator Neural Networks (RNN)* method, which trains a neural network capturing the patterns in the data, and identifies those data points not adhering to those patterns [4].

[1] https://github.com/dwslab/COMMAND

[2] Note that since we expect all base matcher scores to fall in a $[0; 1]$ interval, using geometrical distance measures in that space is feasible.

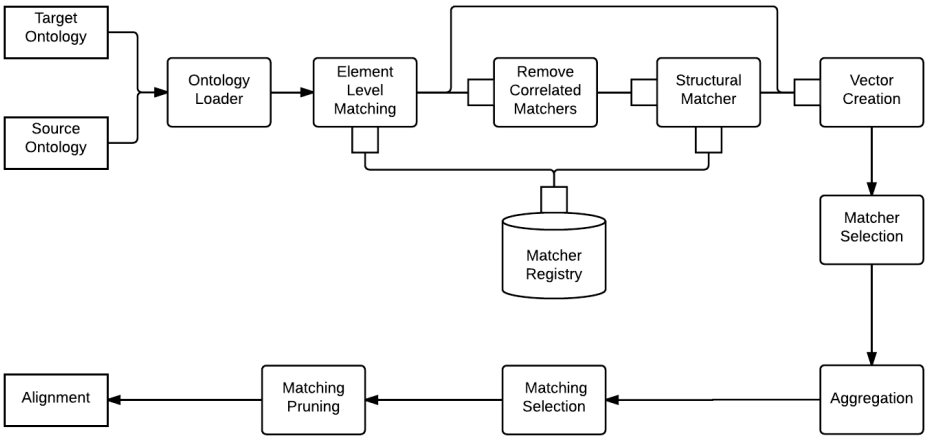


Fig. 1. Overview of the COMMAND pipeline

2.3 Matching Selection and Repair

The result of the previous step is a set of candidates, which does not necessarily form a semantically coherent mapping. After applying a threshold to the results of classes, data and object properties, the mapping may be refined by the *Hungarian method*, a *greedy selection*, or a *fuzzy greedy selection* [2]. Furthermore, logical consistency may be ensured by running the *ALCOMO* mapping post-processing system [6].

3 Evaluation

To evaluate the COMMAND approach, we use the *benchmark*, *conference*, and *anatomy* of the Ontology Alignment Evaluation Initiative (OAEI) 2014 [3].

We compare the results of COMMAND to three baselines. *Single best global* refers to the single base matcher that performs best on the given test case (i.e., conference, benchmark, and anatomy), using the optimal global threshold. *Majority vote* performs a voting across all base matchers, again using the best global threshold. *Single best local* selects the best base matcher for each problem.[3]

Furthermore, we compare COMMAND to the contestants of the OAEI 2014 initiative. To make that comparison fair, we use one global parameter set for each variant across all three OAEI datasets, instead of per dataset settings.

Tables 1, 2, and 3 depict the results of COMMAND on the OEAI datasets, once with and once without the use of ALCOMO. For anatomy, we restrict ourselves to the CBLOF variant and a subset of eight element-level matchers due to reasons of runtime. Except for the *Single best local* baseline (which is informative and not a baseline that can actually be implemented), COMMAND outperforms all baselines. When comparing COMMAND to the results of OAEI

[3] Note that in practice, it would not be possible to implement a matcher like *Single best local*. We only report it for informative purposes.

Table 1. Results on the OAEI biblio benchmark dataset. The table reports macro average recall, precision, and F-measure, with micro average values in parantheses.

Approach	without ALCOMO			with ALCOMO		
	Precision	Recall	F1	Precision	Recall	F1
Single best global	.754 (.733)	.557 (.521)	.641 (.609)	.779 (.761)	.548 (.521)	.644 (.619)
Majority vote	.510 (.472)	.570 (.544)	.538 (.505)	.524 (.487)	.463 (.443)	.491 (.464)
Single best local	.788 (.718)	.632 (.616)	.702 (.663)	.835 (.798)	.610 (.584)	.705 (.674)
CBLOF + PCA	.833 (.983)	.444 (.470)	.579 (.636)	.832 (.981)	.432 (.457)	.568 (.624)
CBLOF + RC	.844 (.982)	.466 (.461)	.600 (.627)	.844 (.982)	.457 (.449)	.593 (.617)
k-NN + PCA	.868 (.977)	.547 (.550)	**.672** (.704)	.871 (.975)	.480 (.459)	**.619 (.624)**
k-NN + RC	.847 (.967)	.549 (.556)	.666 (**.706**)	.835 (.984)	.463 (.442)	.596 (.610)
RNN + PCA	.881 (.991)	.466 (.443)	.610 (.612)	.859 (.965)	.324 (.253)	.470 (.401)
RNN + RC	.877 (.988)	.470 (.448)	.612 (.616)	.877 (.987)	.471 (.450)	.613 (.618)

Table 2. Results on the OAEI conference dataset. The table reports macro average recall, precision, and F-measure, with micro average values in parantheses.

Approach	without ALCOMO			with ALCOMO		
	Precision	Recall	F1	Precision	Recall	F1
Single best global	.641 (.784)	.591 (.611)	.615 (.687)	.640 (.783)	.591 (.611)	.615 (.686)
Majority vote	.874 (.949)	.537 (.552)	.665 (.698)	.874 (.949)	.537 (.552)	.665 (.698)
Single best local	.651 (.795)	.602 (.625)	.626 (.700)	.650 (.793)	.602 (.625)	.625 (.699)
CBLOF + PCA	.693 (.678)	.636 (.613)	.663 **(.644)**	.737 (.715)	.625 (.600)	**.676 (.652)**
CBLOF + RC	.702 (.693)	.607 (.577)	**.651** (.630)	.761 (.752)	.588 (.557)	.663 (.640)
k-NN + PCA	.718 (.712)	.572 (.534)	.636 (.610)	.797 (.782)	.557 (.518)	.656 (.623)
k-NN + RC	.710 (.702)	.574 (.541)	.635 (.611)	.781 (.769)	.530 (.492)	.631 (.600)
RNN + PCA	.829 (.815)	.528 (.492)	.645 (.613)	.748 (.699)	.617 (.587)	**.676** (.638)
RNN + RC	.820 (.805)	.527 (.489)	.641 (.608)	.819 (.804)	.524 (.485)	.639 (.605)

2014, we can find that the system, using CBLOF and PCA, and alignment repair with ALCOMO, would score on rank on a shared fifth rank (with XMap2) for the benchmark track, on rank four for the conference track (between LogMap-C and XMap), and on rank six (between LogMap-C and MaasMatch) for the anatomy track.

The runtime of *COMMAND* is assessed by measuring the time of a complete end-to-end pipeline execution. The general time complexity of *COMMAND* is quadratic to the size of the input ontologies. Additionally, the time consumption of the individual steps is measured. The results are depicted in table 4.

4 Conclusion and Outlook

In this paper, we have introduced a novel approach using anomaly detection for combining the results of different ontology matchers into a final aggregated matching score.

Overall, *COMMAND* performs an efficient *matcher selection* that only considers matchers that contribute to the final result, and uses *anomaly detection* as an unsupervised method for aggregating base matcher results. It is superior

Table 3. Results on the OAEI anatomy dataset.

Approach	without ALCOMO			with ALCOMO		
	Precision	Recall	F1	Precision	Recall	F1
Single best local/global	.920	.773	.840	.918	.740	**.820**
Majority vote	.932	.606	.735	.931	.597	.727
CBLOF + PCA	.892	.728	**.801**	.911	.741	.817
CBLOF + RC	.839	.664	.742	.832	.725	.775

Table 4. Average runtime in seconds of COMMAND

Dataset	Ø total	Ø t vector creation	Ø t aggregation	Ø t extraction
Conference	69.267	53.580	15.683	0.004
Benchmarks	52.880	44.026	8.850	0.004
Anatomy	18,746.510	11,595.601	5,922.478	1,228.431

to a simple majority vote baseline and performs in the range of state of the art matching tools. Furthermore, the possibility to use principal component analysis for feature space transformation also allows for implicitly computing relevant linear combinations of matcher scores.

The evaluation has been carried out on three OAEI datasets. For *conference* and *benchmarks*, the system achieved competitive performances in comparison to other OAEI participants. The results on the *anatomy* track showed that, since only a reduced configuration could be used with sub-optimal results, that more memory-efficient implementations are still required for fully exploiting the capabilities of COMMAND.

Furthermore future work will include the inclusion of other anomaly detection approaches, like angle-based methods, as well as other score normalization methods.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41(3) (2009)
2. Do, H.H., Rahm, E.: Coma: A system for flexible combination of schema matching approaches. In: Proceedings of the 28th International Conference on Very Large Data Bases. pp. 610–621. VLDB '02, VLDB Endowment (2002)
3. Dragisic, Z.e.a.: Results of theontology alignment evaluation initiative 2014. In: International Workshop on Ontology Matching. pp. 61–104 (2014)
4. Hawkins, S., He, H., Williams, G., Baxter, R.: Outlier detection using replicator neural networks. In: Data warehousing and knowledge discovery, pp. 170–180. Springer (2002)
5. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognition Letters 24(9), 1641–1650 (2003)
6. Meilicke, C.: Alignment incoherence in ontology matching. Ph.D. thesis (2011)
7. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. Knowledge and Data Engineering, IEEE Transactions on 25(1), 158–176 (Jan 2013)

User Involvement in Ontology Matching Using an Online Active Learning Approach

Booma Sowkarthiga Balasubramani, Aynaz Taheri, and Isabel F. Cruz

ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
{bbalas3,ataher2,ifcruz}@uic.edu

Abstract. We propose a semi-automatic ontology matching system using a hybrid active learning and online learning approach. Following the former paradigm, those mappings whose validation is estimated to lead to greater quality gain are selected for user validation, a process that occurs in each iteration, following the online learning paradigm. Experimental results demonstrate the effectiveness of our approach.

1 Introduction

The result of performing ontology matching is a set of mappings between concepts in the *source ontology* and concepts in the *target ontology*. This set is called an *alignment*. The *reference alignment* or *gold standard* is (an approximation of) the set of correct and complete mappings built by domain experts. We consider a semi-automatic ontology matching approach, whereby the mappings are first determined using automatic ontology matching methods, which we call *matchers*, followed by user validation.

We use six of the matchers of the AgreementMaker ontology matching system [3], including the Linear Weighted Combination (LWC) matcher, which performs a weighted combination of the results of the other five matchers, using weights that are automatically determined using a quality metric [4].

We train a classifier and modify the weights of the LWC matcher using an iterative approach, following the on-line learning paradigm. At each iteration, user validation is sought for those candidate mappings that can potentially contribute the most to the quality of the final alignment, following the active learning paradigm. The process continues until there is no significant improvement in F-Measure. We describe this process in Section 2. Experimental results are obtained using the ontology sets from the Ontology Alignment Evaluation Initiative (OAEI) and comparison is made with the results of other systems in Section 3. We discuss related work in Section 4, and conclude with Section 5.

2 Proposed System

After the source and target ontologies are loaded into AgreementMaker, the following steps are executed in sequence:

Automatic matching algorithms execution The following matchers are executed individually and their results are stored in the corresponding similarity matrices: the Advanced Similarity Matcher (ASM) [5], the Parametric String-based

Matcher (PSM) [4], the Lexical Similarity Matcher (LSM) [5], the Vector-based Multi-word Matcher (VMM) [4], and the Base Similarity Matcher (BSM) [5].

Linear weighted combination The Linear Weight Combination (LWC) matcher [6] linearly combines the similarity matrices of the other five automatic matchers using weights determined by the local confidence quality metric, which estimates the quality of the scores produced by each matcher. The new score for each mapping is stored in the LWC matrix. It is up to the selection phase to output only those mappings that are in the final alignment, taking into account the desired cardinality of the mappings (e.g., one-to-one) [4].

Candidate mapping selection Candidate mappings to be presented to the users for validation are based on the combination of the following three criteria: (1) Disagreement-based Top-k Mapping [6], which measures the level of similarity among the five scores, one for each of the matchers considered. If the matchers mostly agree on the scores, then the disagreement is low, but it is high when the matchers disagree on the scores; (2) Cross Count Quality (CCQ), which counts, for a score, the number of non-zero scores in the row and column of that score in the LWC matrix [2]. The count is normalized by the maximum sum of the scores per column and row in the whole matrix; (3) Similarity Score Definiteness (SSD), which is a quality metric that ranks mappings in increasing order of their score [2]. It evaluates how close the score associated with a mapping is to the maximum and minimum possible scores (1 and 0).

User validation The result of this step is a label that has value 1 if the mapping is correct and 0 if the mapping is incorrect. For each iteration, users validate a set of candidate mappings. The validation of each mapping is called an *interaction* by others [7]. There can be any number of interactions per iteration, that is, users can be presented with any number of mappings to validate at a time.

Classification We use a logistic regression classifier, which considers the parametric distribution $P(Y|X)$ where Y is the discrete-valued user label (1 or 0) and the feature vector $X = \langle X_1, \ldots, X_n \rangle$ is the signature vector [6] with n scores computed for a mapping by n individual matchers, and estimates the parameter that is the vector of weights $W = \langle w_1, \ldots, w_n \rangle$ of the LWC matcher. The logistic regression model is based on the following probabilities:

$$P(Y = 1|X) = \frac{1}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}, P(Y = 0|X) = \frac{e^{w_0 + \sum_{i=1}^{n} w_i X_i}}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}$$

W is updated during the iterative process by taking the partial derivative of the log likelihood function with respect to each component, w_i. The recursive rule for the update is as follows, where α is the learning rate that determines how fast or slow the weights will converge to their optimal values [10]:

$$W \leftarrow W + \alpha \sum_{i=1}^{m} X^i (Y^i - g(W^T X^i))$$

3 Experimental Evaluation

We use the 2014 OAEI Conference Track ontology sets and their reference alignments to simulate the user validation. The baseline is the F-Measure obtained

automatically by the AgreementMaker matchers. Table 1 depicts the average F-Measure after 20 iterations using the three candidate selection criteria individually or in combination with one another. The top performer is the Disagreement-based Top-k Mapping Selection criteria.

	1	2	3	4	5	6	7
Candidate Mapping Selection Strategy	48.08	52.45	60.43	51.42	48.91	52.47	53.18
Baseline (Before User Feedback)	51.8	51.8	51.8	51.8	51.8	51.8	51.8

Strategies: 1. CCQ 2. SSD 3. Disagreement 4. CCQ + SSD 5. CCQ + Disagreement 6. SSD + Disagreement 7. CCQ + SSD + Disagreement

Table 1: Average F-Measure for 20 iterations (123 interactions/iteration).

Matcher	F-Measure with User Feedback	F-Measure w/o User Feedback	F-Measure gain	Relative Number of Interactions
AML	0.801	0.730	0.071	0.497
LogMap	0.729	0.680	0.049	0.391
HerTUDA	0.582	0.600	-0.018	0.996
WeSeE	0.473	0.610	-0.137	0.447
Our Approach	0.604	0.518	0.086	0.470

Table 2: Comparison with the 2014 OAEI Interactive Track results.

Our approach has an average F-Measure gain of 8.6% and an average F-Measure of 60.4%. This is a considerable improvement as we started from an average F-Measure of 51.8%, which was obtained using the automatic matchers along with LWC. Table 2 compares our results with those obtained by other systems that participated in the 2014 OAEI Interactive Track. It performs better than HerTUDA and WeSeE (with F-Measure values of 58.2% and 47.3%, respectively). The F-Measure gain of AML [9] is 7.1% and of LogMap is 4.6%, therefore our approach has the highest F-Measure gain. The table also shows the relative number of interactions, which is the average number of interactions per pair of ontologies divided by the size of the reference alignment for that pair. Our approach shows better improvement in F-Measure with fewer number of interactions when compared to AML that has the highest F-Measure.

Figure 1 shows the effect of the total number of interactions on the F-Measure in our approach. Here, the total number of interactions represent the sum of the number of interactions in each of the 21 reference alignments in the Conference Track dataset (one for each pair of ontologies) up to 123 interactions. The Disagreement-based Top-k Mapping Selection performs better than the other candidate selection strategies. SSD and the combination of SSD+CCQ+ Disagreement have the next highest average F-Measure.

4 Comparison with Related Work

We divide previous work into two categories depending on whether feedback from single or multiple users is considered.

Single user A previous approach that uses AgreementMaker performs updates in the LWC matrix based on user feedback [6], but does not use a classifier to adjust

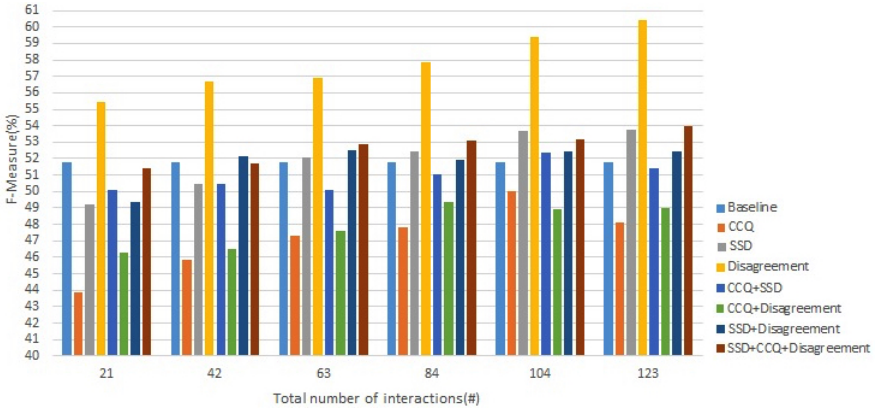


Fig. 1: F-Measure gain as a function of the number of interactions.

the LWC weights. Another method uses logistic regression to learn an optimal combination of both lexical and structural similarity metrics [8]. Compared to our approach, it uses different similarity metrics, candidate selection strategies, and techniques to customize weights for different matching strategies. Another system aggregates similarity measures with the help of self-organizing maps and incorporates user feedback for refining self-organizing map outcomes [11]. There is an active learning approach where the user validation is propagated according to the ontology structure [13]. Another approach makes use of the parameterization of matchers [12]. It uses example mappings to automatically determine a suitable parameter setting for each matcher, based on those examples. However, in our approach, the LWC uses five of the already existing matchers with the same configuration as in AgreementMaker.

Multiple users We discuss two approaches. The first one uses a pay-as-you-go approach and propagates the (possibly faulty) user validation input to similar mappings [2]. In the second approach, a multi-user feedback method that attempts to maximize the benefits that can be drawn from user feedback, by managing it as a first class citizen [1]. None of these approaches uses a classifier.

5 Conclusions and Future Work

In this paper, we have proposed an effective semi-automatic ontology matching approach that combines active learning with online learning. Our experimental evaluation demonstrate that a considerable improvement in F-Measure can be achieved over the base case. Clearly, a combination of user feedback with learning is fertile ground for future research, where the scalability of the methods to large and very large ontologies and the use of a variety of classifiers and of candidate selection strategies would be some of the topics to investigate.

Acknowledgments

This research was partially supported by NSF Awards IIS-1143926, IIS-1213013, and CCF-1331800.

References

1. Belhajjame, K., Paton, N.W., Fernandes, A.A.A., Hedeler, C., Embury, S.M.: User Feedback as a First Class Citizen in Information Integration Systems. In: CIDR Conference on Innovative Data Systems Research. pp. 175–183 (2011)
2. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-As-You-Go Multi-User Feedback Model for Ontology Matching. In: International Conference on Knowledge Engineering and Knowledge Management (EKAW), pp. 80–96. Springer (2014)
3. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB 2(2), 1586–1589 (2009)
4. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: ISWC International Workshop on Ontology Matching (OM). CEUR Workshop Proceedings, vol. 551, pp. 49–60 (2009)
5. Cruz, I.F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Palandri Antonelli, F., Keles, U.C.: Using AgreementMaker to Align Ontologies for OAEI 2010. In: ISWC International Workshop on Ontology Matching (OM). CEUR Workshop Proceedings, vol. 689, pp. 118–125 (2010)
6. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: IEEE International Conference on Data Engineering (ICDE). pp. 1321–1324 (2012)
7. Dragisic, Z., Eckert, K., Euzenat, J., Faria, D., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A.O., Lambrix, P., Montanelli, S., Paulheim, H., Ritze, D., Shvaiko, P., Solimando, A., dos Santos, C.T., Zamazal, O., Grau, B.C.: Results of the Ontology Alignment Evaluation Initiative 2014. In: ISWC International Workshop on Ontology Matching (OM). pp. 61–104. CEUR Workshop Proceedings (2014)
8. Duan, S., Fokoue, A., Srinivas, K.: One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback. In: International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 6496, pp. 177–192. Springer (2010)
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight Ontology Matching System. In: International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE). pp. 527–541. Springer (2013)
10. Halloran, J.: Classification: Naive Bayes vs Logistic Regression. Tech. rep., University of Hawaii at Manoa EE 645 (2009)
11. Jirkovskỳ, V., Ichise, R.: Mapsom: User Involvement in Ontology Matching. In: Joint International Semantic Technology Conference (JIST), pp. 348–363. Springer (2014)
12. Ritze, D., Paulheim, H.: Towards an Automatic Parameterization of Ontology Matching Tools Based on Example Mappings. In: ISWC International Workshop on Ontology Matching (OM). pp. 37–48 (2011)
13. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively Learning Ontology Matching via User Interaction. In: International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 5823, pp. 585–600. Springer (2009)

ADOM: Arabic Dataset for Evaluating Arabic and Cross-lingual Ontology Alignment Systems

Abderrahmane Khiat[1], Moussa Benaissa[1], and Ernesto Jiménez-Ruiz[2]

[1]LITIO Laboratory, University of Oran1 Ahmed Ben Bella, Oran, Algeria
[2]Department of Computer Science, University of Oxford, United Kingdom

Abstract. In this paper, we present ADOM, a dataset in Arabic language describing the conference domain. This dataset was created for two purposes (1) analysis of the behavior of matchers specially designed for Arabic language, (2) integration with the multifarm dataset of the Ontology Alignment Evaluation Initiative (OAEI). The multifarm track evaluates the ability of matching systems to deal with ontologies described in different natural languages. We have tested the ADOM dataset with the LogMap ontology matching system. The experiment shows that the ADOM dataset works correctly for the task of evaluating cross multilingual ontology alignment systems.

1 Introduction

Ontology alignment is defined as the identification process of semantic correspondences between entities of different ontologies in order to ensure the semantic interoperability [1]. However, the automatic identification of correspondences between ontologies is very difficult due to (a) their conceptual divergence [8], and (b) to the use of different naming conventions or languages. In the literature there are several systems that deal with the (semi) automatic alignment of ontologies [1, 12, 11]. These systems are (typically) primarily based on the lexical similarity of the entity labels. Matching ontologies in different languages is challenging due to misinterpretations during the translation process. Ontologies in Arabic language brings even more challenges due to special features of the language. Among the reasons that make ontology alignment in Arabic language very difficult we can quote [6]:

1. The Arabic script (no short vowels and no capitalization).
2. Explosion of ambiguity (in average 2.3 per word in other languages to 19.2 in Arabic) by Buckwalter (2004) [5].
3. Complex word structure, for example the sentence ورأيتهم can be translated in English language as and I saw them.
4. The problem of Normalization, for example آ ، إ ، أ ، ا → ا i.e. losing distinction أن ، إن ، آن
5. The Arabic language is one of the pro-drop languages, i.e. languages that allow speakers to omit certain classes of pronouns

Table 1: Top systems in the multifarm track

OAEI	Top Systems	Precision	F-measure	Recall
2012	YAM++	0.50	0.40	0.36
2013	YAM++	0.51	0.40	0.36
2014	AML	0.57	0.54	0.53
2014	LogMap	0.80	0.40	0.28
2014	XMap	0.31	0.35	0.43

In this paper, we present ADOM, a dataset in Arabic language describing the conference domain. We have created this dataset by translating and improving all ontologies of the conference track [13] of the OAEI campaign. We summarize below the objectives of the developed dataset: (1) Analysis and evaluation of the behaviour of matchers designed for Arabic language. Here, the real questions are: (a) could the state of the art systems handle efficiently the ontologies described in Arabic language? (b) Are external knowledge resources for Arabic language available such as WordNet? (2) Integration with the multifarm track [14] of the OAEI campaign.[1] The multifarm track evaluates the ability of matching systems to deal with ontologies described in different natural languages. The question here, concerns to the performance of the translator used to align multilingual ontologies?

The rest of the paper is organized as follows. First, in Section 2, we discuss the top systems that participated in the last editions of the multifarm track. In section 3 we describe the ADOM dataset. Section 4 contains the experiment results. Finally, some concluding remarks and future work are presented in Section 5.

2 Related Work

In this section we discuss the main ontology matching systems that have participated in the multifarm track. Most of such systems use a translation tool to deal with the cross-lingual ontology alignment. The XMap system [2] uses an automatic translation for obtaining correct matching pairs in multilingual ontology matching. The translation is done by querying Microsoft Translator for the full name. The AML system [4] uses an automatic translation module based on Microsoft Translator. The translation is done by querying Microsoft Translator for the full name (rather than word-by-word). To improve performance, AML stores locally all translation results in dictionary files, and queries the Translator only when no stored translation is found. The LogMap system [10] that participated in the OAEI 2014 campaign used a multilingual module based on Google translate [3]; however the new version of the LogMap system uses both Microsoft and Google translator APIs [9]. The YAM++ system [7] uses a multilingual translator based on Microsoft Bing to translate the annotations to English. Table 1 summarizes the results of the top systems in the multifarm track.

[1] ADOM has already been integrated within the OAEI 2015 multifarm dataset: `http://oaei.ontologymatching.org/2015/multifarm/index.html`

3 The ADOM Dataset

The dataset is constituted of seven ontologies in Arabic language. These ontologies describe the conference domain and are based on the ontologies of the OAEI conference track [13]. We justify the proposal of our dataset by the following points: (1) The OAEI campaign, which is the most known evaluation campaign for testing the performance of ontology matching systems, lacked a test case involving ontologies in Arabic language. (2) To the best of our knowledge, no such dataset exists yet in Arabic language.[2] (3) Furthermore, there are several contexts such as Web information retrieval where the ontology matching systems are needed both in inter-multilingual ontologies and intra-Arabic ontologies.

We have developed our dataset relying on the conference and multifarm tracks of the OAEI. In order to develop the Arabic ontologies and reference alignments for the ADOM dataset we proceeded as follows.

3.1 Step 1: Translation of Ontology Entities

In this step, we have identified the concepts, object and data-type properties of the ontologies, for example we can list the concept "البحث (paper)", data-type property "لديه اسم (has name) and object property "لديه موقع على رابط (has website at URL), etc. We have semi-automatically translated the ontologies in English and French by considering the context of the ontologies (i.e., the conference domain). For example, if we translate simply the concept "paper" we get "ورقة" in Arabic language but "ورقة" is not the correct concept if we consider the context of conference and some information from conference websites in Arabic language. Then the correct concept of "paper" becomes "البحث".

3.2 Step 2: Generation of Reference Alignments

We have reused the available reference alignments among the ontologies in the multifarm track to generate the new reference alignments for ADOM. For example, in the reference alignment for ontologies in Arabic language, we can list the concept "الحدث (event)" of the ontology Confof is equivalent to the concept "نشاط (activity)" of the ontology Iasted. In the reference alignment for ontologies in Arabic and French languages, we can list the concept "éditeur (Editor)" of the ontology conference is equivalent to the concept "المحرر" of the ontology Cmt.

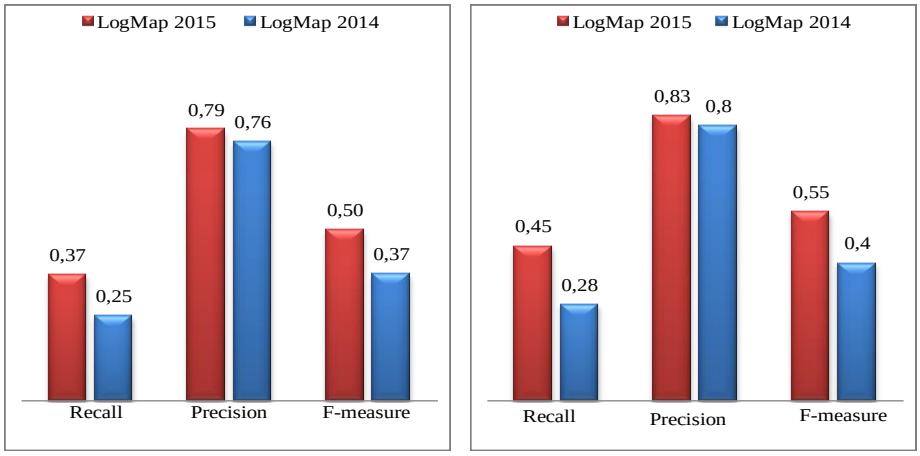
3.3 Step 3: Validation by a Linguistic Expert

Our dataset was validated by a linguistic expert with regard to the translation of concepts and properties. Furthermore we also checked the correctness of the new reference alignments.

[2] Note that, in the literature one can find datasets in Arabic language applied to other domains different from Ontology Matching (e.g. [15, 16])

4 Experimental Study

In order to evaluate the ADOM dataset, we have used the LogMap system which is one of the top ontology alignment systems on multifarm track (see Table 1). The purpose of this evaluation is to show that the ADOM dataset is suitable to test ontology matching systems that implement multilingual support.



(a) LogMap results on ADOM dataset (b) LogMap System on Multifarm Track

Fig. 1: LogMap 2014 and 2015 results on ADOM and in all multifarm dataset.

We have tested the ADOM dataset with two versions of LogMap system. The first version, which has participated in the OAEI 2014, uses the Google translator API. The second, which aims at participating in the OAEI 2015, uses both the Microsoft and Google translators APIs. Figure 1 summarizes the average results, in terms of precision, recall and F-measure, obtained by LogMap on the ADOM dataset (Fig. 1a) and on all multifarm tests (Fig. 1b). We can appreciate that, on average, the ADOM dataset brings an additional complexity to the multifarm track, with regard the results obtained by LogMap. Note that, we aim at obtaining a more comprehensive evaluation during the OAEI 2015 evaluation campaign to confirm this fact.

5 Conclusion

In this paper we have presented ADOM, a new dataset in Arabic language describing the conference domain. This dataset has been created for two purposes 1) studying and developing specific ontology alignment methods to align ontologies in Arabic language, 2) evaluating the ability of state of the art ontology matching systems to deal with ontologies in Arabic. The experimental study shows that ADOM dataset is suitable in practice. Furthermore, ADOM has already been integrated within the multifarm dataset and it will be evaluated in the OAEI 2015 campaign.

As future challenges, we aim at (1) developing a large corpus of ontologies and dictionaries for the Arabic language, (2) adapting state of the art NLP tools to align ontologies in Arabic language, (3) improving the state of the art translators dedicated to the Arabic language.

References

1. J. Euzenat and P. Shvaiko, "Ontology Matching", Springer-Verlag, Heidelberg, 2013.
2. W. Djeddi and M. T.Khadir, "XMap++ results for OAEI 2014". In Proceedings of the 9th International Workshop on Ontology Matching ISWC 2014, pp. 163169, Italy, 2014.
3. E. Jiménez-Ruiz, B. C. Grau, W Xia, A. Solimando, X. Chen, V. Cross, Y. Gong, S. Zhang and A. Chennai-Thiagarajan, "LogMap family results for OAEI 2014". In Proceedings of the 9th Workshop on Ontology Matching ISWC 2014, pp. 126-134, Italy, 2014.
4. D. Faria, C. Martins, A. Nanavaty, A. Taheri, C. Pesquita, E. Santos, I. F. Cruz and F. M. Couto, "AgreementMakerLight results for OAEI 2014". In Proceedings of the 9th Workshop on Ontology Matching ISWC 2014, pp. 105-112, Italy, 2014.
5. T. Buckwalter, "Arabic Morphological Analyzer Version 2.0". LDC catalog number LDC2004L02, 2004.
6. A. Farghaly, "Arabic NLP: Overview, state of the art, challenges and opportunities", In The International Arab Conference on Information Technology, ACIT2008, Tunisia, 2008.
7. D. Ngo and Z. Bellahsene, "YAM++ results for OAEI 2013", In Proceedings of the 8th Workshop on Ontology Matching ISWC 2013, pp. 211-218, Australia, 2013.
8. P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou and S. Tessaris "Specification of a Common Framework for Characterizing Alignment", Deliverable 2.2.1, Knowledge Web NoE, Technical Report, Italy, 2004.
9. E. Jiménez-Ruiz et al. "LogMap family results for OAEI 2015". In Proceedings of the 10th Workshop on Ontology Matching ISWC 2015, pp., USA, 2015.
10. E. Jiménez-Ruiz, Bernardo Cuenca Grau, Yujiao Zhou and Ian Horrocks. "Large-Scale Interactive Ontology Matching: Algorithms and Implementation". In: ECAI. 2012.
11. Z. Dragisic, K. Eckert, J. Euzenat, D. Faria, A. Ferrara, R. Granada, V. Ivanova, E. Jiménez-Ruiz, A. O. Kempf, P. Lambrix, S. Montanelli, H. Paulheim, D. Ritze, P. Shvaiko, A. Solimando, C. Trojahn-dos-Santos, O. Zamazal and B. Cuenca Grau, "Results of the Ontology Alignment Evaluation Initiative 2014", 9th Workshop on Ontology Matching, 2014.
12. B. Cuenca Grau, Z. Dragisic, K. Eckert, J. Euzenat, A. Ferrara, R. Granada, V. Ivanova, E. Jiménez-Ruiz, A. Oskar Kempf, P. Lambrix, A. Nikolov, H. Paulheim, D. Ritze, F. Scharffe, P. Shvaiko, C. Trojahn dos Santos, O. Zamazal, "Results of the Ontology Alignment Evaluation Initiative 2013". 8th Workshop on Ontology Matching, 2013.
13. O. Svab, V. Svatek, P. Berka, D. Rak and P. Tomasek, "OntoFarm: Towards an Experimental Collection of Parallel Ontologies", In: Poster Track of ISWC 2005, Galway, 2005.
14. C. Meilicke, R. Garca-Castro, F. Freitas, WR. Van Hage, E. Montiel-Ponsoda, R.R. De Azevedo, H. Stuckenschmidt, O. vb-Zamazal, V. Svtek and A. Tamilin, "MultiFarm: A benchmark for multilingual ontology matching". Web Semant. Sci. Serv. Agents World Wide Web. Vol. 15, pp. 6268, 2012.
15. I. Bounhas, B. Elayeb, F. Evrard, and Y. Slimani, "ArabOnto: Experimenting a new distributional approach for Building Arabic Ontological Resources". In International Journal of Metadata, Semantics and Ontologies, Inder-science, Vol. 6, No. 2, pp. 81-95, 2011.
16. O. Ben Khiroun, R. Ayed, B. Elayeb, I. Bounhas, N. Bellamine Ben Saoud and F. Evrard, "Towards a New Standard Arabic Test Collection for Mono- and Cross-Language Information Retrieval", In the Proceedings of 19th International Conference on Application of Natural Language to Information Systems (NLDB), 2014.

Ontology Matching for Big Data Applications in the Smart Dairy Farming Domain

Jack P.C. Verhoosel, Michael van Bekkum and Frits K. van Evert

TNO Connected Business, Soesterberg, The Netherlands
{jack.verhoosel,michael.vanbekkum}@tno.nl
Wageningen UR, Wageningen, The Netherlands
frits.vanevert@wur.nl

Abstract. This paper addresses the use of ontologies for combining different sensor data sources to enable big data analysis in the dairy farming domain. We have made existing data sources accessible via linked data RDF mechanisms using OWL ontologies on Virtuoso and D2RQ triple stores. In addition, we have created a common ontology for the domain and mapped it to the existing ontologies of the different data sources. Furthermore, we verified this mapping using the ontology matching tools HerTUDA, AML, LogMap and YAM++. Finally, we have enabled the querying of the combined set of data sources using SPARQL on the common ontology.

1 Background and context

Dairy farmers are currently in an era of precision livestock farming in which information provisioning for decision support is becoming crucial to maintain a competitive advantage. Therefore, getting access to a variety of data sources on and off the farm that contain static and dynamic individual cow data is necessary in order to provide improved answers on daily questions around feeding, insemination, calving and milk production processes.

In our SmartDairyFarming project, we have installed sensor equipment to monitor around 300 cows each at 7 dairy farms in The Netherlands. These cows have been monitored during the year 2014 which has generated a huge amount of sensor data on grazing activity, feed intake, weight, temperature and milk production of individual cows stored in databases at each of the dairy farms. The amount of data recorded per cow is at least 1MB of sensor values per month, which adds up to 3.6GB of data per dairy farm per year. In addition, static cow data is available in a data warehouse at the national milk registration organization, including date of birth, ancestors and current farm. Finally, another existing data source contains satellite information on the amount of biomass in grasslands in the country that is important for measuring the feed intake of cows during grazing.

We focused on decision support for the dairy farmer on feed efficiency in relation to milk production. Thus, the big data analysis question is: “How much feed did an individual cow consume in a certain time period at a specific grassland parcel and how does this relate to the milk production in that period?”.

2 Ontology matching approach

We selected one of the dairy farms (DairyCampus) and created with TopBraid composer a small ontology with 12 concepts that covers among others the grasslands

of a farm and grazing periods of cows. This ontology contains the concept "perceel" which is Dutch for parcel. In addition, we selected the data source with satellite information about biomass in grasslands (AkkerWeb, www.akkerweb.nl). This data source already had an ontology defined with 15 concepts that contains the concept "plot" which is similar to parcel but with different properties. Furthermore, we created with TopBraid composer a common ontology for the domain with 28 concepts on feed efficiency (see **Fig. 1**).

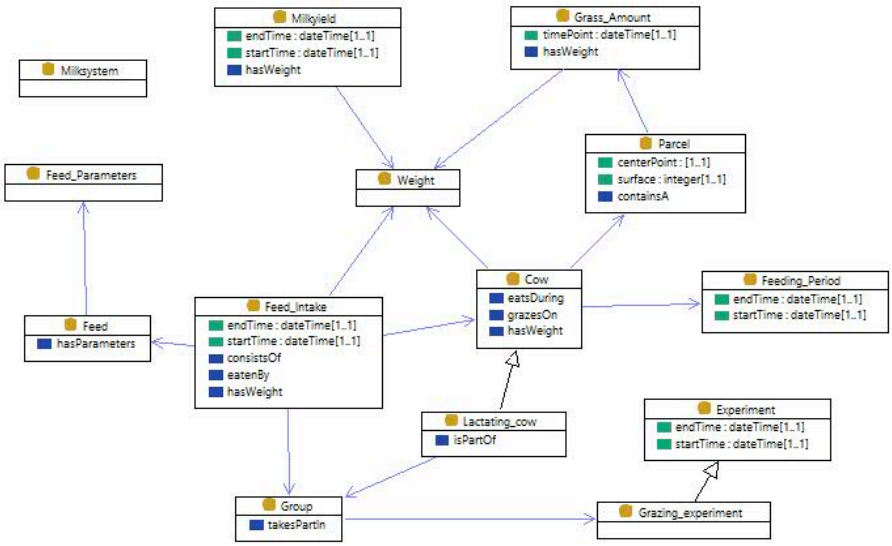


Fig. 1. Common ontology excerpt for feed efficiency in dairy farming.

The challenge was to find a match between the concepts and properties in the common ontology and both specific DairyCampus and Akkerweb ontologies, especially regarding the concepts "parcel", "perceel" and "plot".

We have initially created manual mappings between classes and properties in TopBraid using rdfs:subClassOf and owl:equivalentProperty relations. Based on relatively few and simple matches we created initial alignments between properties and classes (see **Fig. 2**).

Use of a matching tool or system however, provides us with opportunities to verify our current findings and better support our efforts in finding alignments between the other concepts in our ontologies. We used a literature survey of matching techniques and supporting matching systems in [1] to identify both a suitable matching technique and find tools supporting that technique. We consider language-based matching as the appropriate type of matching since it focuses on syntactic element-level natural language processing of words.

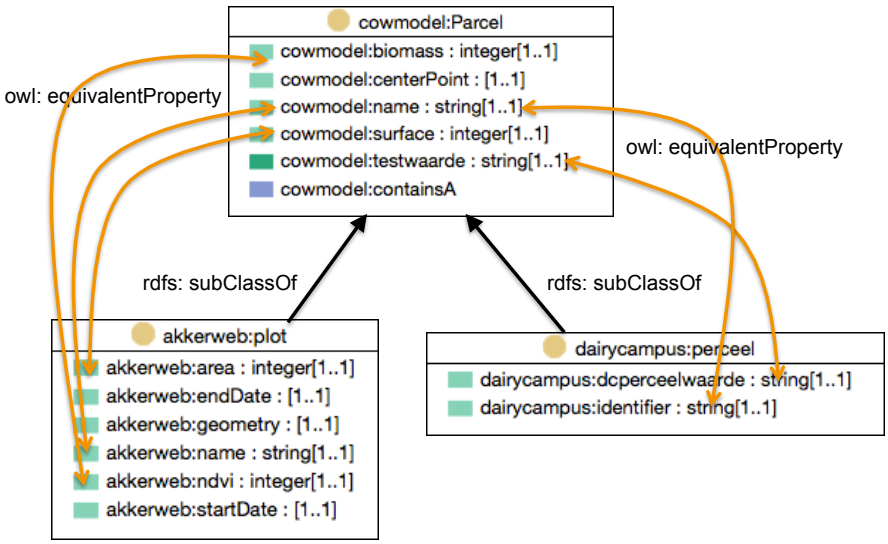


Fig. 2. Mapping of classes and properties based on the matching result.

There are numerous tools available that support this specific matching technology, mostly from academic efforts. Some however are no longer in active use, either being outdated or not maintained anymore [2].

We have selected several matching systems that support our requirement of language-based matching: HerTUDA [3,4], AgreementMaker Light (AML) [5], LogMap [6], and YAM++ [7]. We have started to investigate the possibilities of these tools to find alignments of concepts and properties in our ontologies. Initial efforts with the concepts shown in **Fig. 2** have not led to successful matches and alignments yet, however. The HerTUDA, LogMap and YAM++ tools were difficult to install and execute. The AML worked fine, but could not entirely find the relation between "parcel", "perceel" and "plot". Further analysis is required to find out whether this is due to inappropriate matching techniques or to the specific ontologies that we offered to the tool.

3 SPARQL queries and triple stores

In order to show that the mapping of the common ontology to the specific ontologies works properly, we generated in TopBraid a few instances of an Akkerweb plot and a DairyCampus perceel. In addition, we build a simple select query using the common ontology to retrieve all parcels and for each parcel the properties name, biomass, surface and test.

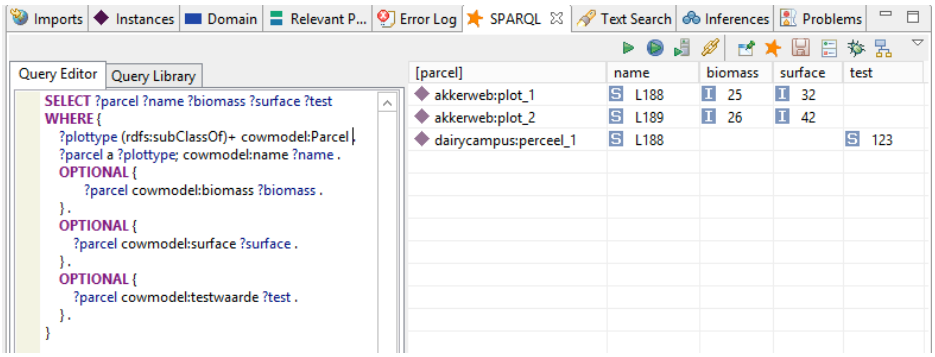


Fig. 3. Select query on common ontology to retrieve all parcels.

The query and its results are shown in **Fig. 3**. As can be seen, the query retrieves both Akkerweb plots and DairyCampus percelen. In addition, Akkerweb contains data about a plot with name "L188" and DairyCampus contains data on a perceel with an identifier "L188". This means that both databases contain the same parcel and the properties can be combined.

The specific ontologies for DairyCampus and Akkerweb formed the basis to generate triples from the relational data sources of DairyCampus and Akkerweb. The triples have been made available via Virtuoso as well as directly from the D2RQ tool (www.d2rq.org). A system that is based on the common ontology can take the big data question to create federated SPARQL queries on the DairyCampus and Akkerweb triple stores using the matched ontologies. As a result, farmers can pose questions in terms of the concepts in the common ontology instead of the detailed and specific concepts of the DairyCampus and Akkerweb data sources.

The farmer can use such a system for decision support purposes on various daily operations, such as which amount of feed to provide to which cow in which period, when to inseminate a specific cow and how to deal with the transition of a cow towards calving.

4 Future work

The approach that is describe in this paper is currently in an experimental phase. We have reached a set-up by filling the triple stores for 3 farms with cow-data of 1 month which adds up to a total of 7 million triples. This needs to be upgraded to all farms with all data from 2014. Thereby, we can test the scalability of our system. In addition, we need to do more detailed analysis of the matching tools that we used and the reasons for not adequately solving the simple matching problem that we proposed.

References

1. Otero-Cerdeira, L., Rodriguez-Martinez, F.J., Gomez-Rodriguez, A.: Ontology matching: A literature review. Journal on Expert Systems with Applications, 949-971 (2015)
2. Ontology matchings tool overview: www.mkbergman.com/1769/50-ontology-mapping-and-alignment-tools/

3. Hertling, S.: Hertuda results for OAEI 2012. In Ontology Matching 2012 workshop proceedings, 141-144 (2012)
4. HerTUDA download: www.ke.tu-darmstadt.de/resources/ontology-matching/hertuda
5. AgreementMakerLight website: somer.fc.ul.pt/aml.php
6. LogMap website: www.cs.ox.ac.uk/isg/tools/LogMap/
7. YAM++ website: www.lirmm.fr/yam-plus-plus