Institut für
Angewandte Informatik
und Formale Beschreibungsverfahren

AIFB

Universität Karlsruhe (TH)

Ontology Mapping – An Integrated Approach

von

Marc Ehrig, York Sure

Bericht 427

April 2004

Forschungsberichte

Herausgeber: A. Oberweis, H. Schmeck, D. Seese, W. Stucky, R. Studer

Institut für
Angewandte Informatik
und Formale Beschreibungsverfahren

AIFB

UNIVERSITÄT KARLSRUHE (TH)

Telefon:

0721-608-4516	**(Prof. Dr. A. Oberweis)**
0721-608-4242	**(Prof. Dr. H. Schmeck)**
0721-608-6037	**(Prof. Dr. D. Seese)**
0721-608-3812	**(Prof. Dr. W. Stucky)**
0721-608-3923	**(Prof. Dr. R. Studer)**

Telefax:
0721-693717

Electronic Mail:
oberweis@aifb.uni-karlsruhe.de
schmeck@aifb.uni-karlsruhe.de
seese@aifb.uni-karlsruhe.de
stucky@aifb.uni-karlsruhe.de
studer@aifb.uni-karlsruhe.de

Postanschrift: Institut AIFB, Universität Karlsruhe (TH), D - 76128 Karlsruhe

Ontology Mapping - An Integrated Approach

Marc Ehrig and York Sure

April 28, 2004

Abstract

Semantic mapping between ontologies is a core issue to solve for enabling interoperability across the Semantic Web. To handle the increasing number of individual ontologies it becomes necessary to develop automatic approaches. In this paper we present such an approach. In brief, we determine similarity through rules which have been manually formulated by ontology experts. The outcomes of these rules are then aggregated to gain one overall result. Several boosting small actions follow. In the end we have pairs of ontological entities mapping to one another. Finally, we evaluate the approach thoroughly with very promising results.

1 Introduction

1.1 Motivation

The Semantic Web community has achieved a good standing within the last years. As more and more people get involved, many individual ontologies are created. Interoperability among different ontologies becomes essential to gain from the power of the Semantic Web. Thus, mapping and merging of ontologies becomes a core question. As one can easily imagine, this can not be done manually beyond a certain complexity, size, or number of ontologies any longer. Automatic or at least semi-automatic techniques have to be developed to reduce the burden of manual creation and maintenance of mappings. Similar questions arise in other knowledge representation domains such as e.g. UML or Entity-Relation-Models .

1.1.1 Scenario

One specific application at Karlsruhe, which requires mapping and merging is derived from the SWAP project (Semantic Web and Peer-to-Peer). The SWAP project[1] wants to enable individuals to keep their own work views and at the same time share knowledge across a peer-to-peer network. For this reason tools are provided for each peer to easily create an own ontology. This ontology represents the view on the local file system, emails, or bookmarks. Through the peer-to-peer network communication between the individual peers becomes possible without relying on a central instance. Formal queries

[1] http://swap.semanticweb.org

are sent around in this network, and peers which know an answer reply to these queries. (Natural language) Examples for such queries could be: "What is the the email address of York?" or "Which documents on Marc's computer are about similarity?". This knowledge can then be integrated into the knowledge repository of the original asking peer. Additionally every peer advertises the topics he has most information on; this expertise is saved by the other peers.
In our scenario mapping becomes necessary for different tasks. Mapping is required every single time a decision is taken on which peer has knowledge about a certain topic, and thus will be addressed with the query. Naturally, a foreign peer can only answer incoming queries, if it can interpret the entities with respect to its own knowledge base. Query rewriting is required [CGL+04]. Finally, the originally asking peer receives answers. When including this information into the own local knowledge base, the new knowledge has to be linked to already existing knowledge. Equal entities have to be identified. Mapping therefore plays a big role in the project SWAP, which was a core motivation for the research presented in this paper.

1.2 Overview

In this paper we present an approach to combine different similarity measures to find mapping candidates between two or more ontologies. **As our hypothesis H we expect better mapping results from intelligent approaches in combining different similarity identifying measures than today's approaches can provide.**

The next section defines and explains general concepts this work is based on: ontology, similarity, and mapping. In section 3 the similarity methods based on rules derived by human experts are introduced. The section 4 presents our approach for combining and integrating these various methods. In section 5 a thorough evaluation is performed showing the strengths of our approach. Finally related work, the next steps, and a conclusion are given.

2 Definitions

In this section our understanding of ontologies is presented. For clarification we also discuss the general meaning of similarity. Additionally follow ideas on how to bring the two worlds together. Our notion of mapping will be presented in the end.

2.1 Ontologies

As the knowledge we want to map is represented in ontologies, we will briefly define this term here.

In philosophy an ontology is "a particular theory about the nature of being or the kinds of existents"[MW]. It also covers the basic question on how to record the real world. [Gru93] discuss a more technical view of ontologies, the most prominent definition being "An ontology is an explicit specification of a conceptualization". A conceptualization refers to an abstract model of some phenomenon in the world by identifying the relevant concept of that phenomenon. Explicit means that the types of concepts used and the constraints on their use are explicitly defined. This definition is often extended by three additional conditions: “An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest”. Formal refers to the fact that the ontology should be machine readable (which excludes for instance natural language). Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted as a group. The reference to a domain of interest indicates that for domain ontologies one is not interested in modelling the whole world, but rather in modelling just the parts which are relevant to the task at hand.

The following short definition describes ontologies as used in our scenario. In the understanding of this paper they consist of schema and metadata.

$$O := (C, H_C, R_C, H_R, I, R_I, A)$$

An ontology O is a tuple consisting of the following. The concepts C of the schema are arranged in a subsumption hierarchy H_C. Relations R_C exist between single concepts. Relations (properties)[2] can also be arranged in a hierarchy H_R. Instances I of a specific concept are interconnected by property instances R_I. Additionally one can define axioms A which can be used to infer knowledge from already existing one. An extended definition can be found in [SEH$^+$03]. Common languages to represent ontologies are RDF(S)[3] or OWL[4], though one should note that each language offers different modelling primitives and, thus, a different level of complexity.

Graph 1 shows a graphical representation of a simple ontology. Human beings and political regions are modelled within a hierarchy. There are relations between the concepts such as a person having a birhtplace in a political region. Finally we show instances of country: Russia, Germany, etc.

2.2 Similarity

We start with a short definition of similarity from Merriam Webster’s Dictionary: *having characteristics in common: strictly comparable.* From our point of view we want to strictly compare two entities to find identity among them. The definition already gives us a hint on how to check for similarity: two entities need common characteristics to be similar. We also give a formal definition of similarity here derived from [Bis95]:

- $sim(x, y) \in [0..1]$
- $sim(x, y) = 1 \rightarrow x = y$: two objects are identical.
- $sim(x, y) = 0$: two objects are different and have no common characteristics.
- $sim(x, x) = 1$: similarity is reflexive.
- $sim(x, y) = sim(y, x)$: similarity is symmetric.[5]

[2] In this paper we treat the words *relation* and *property* as synonyms.
[3] http://www.w3.org/RDFS/
[4] http://www.w3.org/OWL/
[5] We assume symmetry in this paper, although we are aware that it is controversially discussed [MW01].

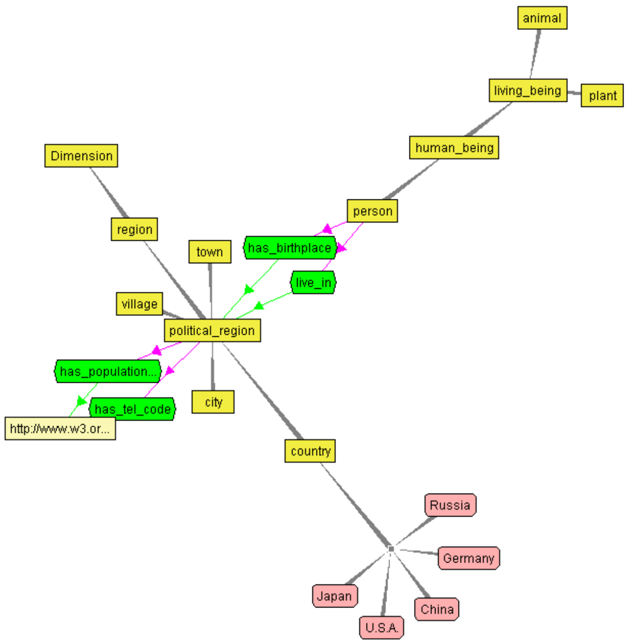

Figure 1: Clipping of an ontology

- similarity and distance are inverse to each other.
- $sim(x, z) \leq (sim(x, y) + sim(y, z))$: The triangular inequation is valid for the similarity measure.[6]

2.3 Similarity for Ontologies

What is the meaning of similarity in the context of ontologies? The basic assumption is that knowledge is captured in an arbitrary ontology encoding. Based on the consistent semantics the coherences modelled within the ontology become understandable and interpretable. From this it is possible to derive additional knowledge such as, in our case, similarity of entities in different ontologies. An example shall clarify how to get from encoded semantics to similarity: by understanding that labels describe entities in natural language one can derive that entities having the same labels are similar. This is not a rule which always holds true, but it is a strong indicator for similarity. Other constructs as subclass relations or type definition can be interpreted similarly.

After the general descriptions of ontologies and similarity we now want to give a formal definition of similarity for ontologies.

- O_i: ontology, with ontology index $i \in \mathbb{N}$
- $sim(x, y)$: similarity function

[6]Please notice that the triangular inequation is not always true [**?**].

- e_{ij}: entities of O_i, with $e_{ij} \in \{C_i, R_i, I_i\}$, entity index $j \in \mathbb{N}$
- $sim(e_{i_1j_1}, e_{i_2j_2})$: similarity function between two entities $e_{i_1j_1}$ and $e_{i_2j_2} (i_1 \neq i_2)$; as shown later this function makes use of the ontologies of the entities compared

A key question remains how the function $sim(e_{i_1j_1}, e_{i_2j_2}, O_{i_1}, O_{i_2})$ can be computed. This will be part of the discussion in this paper.

2.4 Mapping

Due to the wide range of expressions used in this area (merging, alignment, integration etc.), we want to describe our understanding of the term "mapping". We define mapping as cf. [Su02]: "Given two ontologies A and B, mapping one ontology with another means that for each concept (node) in ontology A, we try to find a corresponding concept (node), which has the same or similar semantics, in ontology B and vice verse." Other but similar definitions are given by [DFKO01] or [Sow97]. We want to stick to this definition, more specific we will demand the *same* semantic meaning of two *entities*.

Formally an ontology mapping function can be defined the following way:

- $map : O_{i_1} \rightarrow O_{i_2}$
- $map(e_{i_1j_1}) = e_{i_2j_2}$, if $sim(e_{i_1j_1}, e_{i_2j_2}) > t$ with t being the threshold
  entity $e_{i_1j_1}$ is mapped onto $e_{i_2j_2}$; they are semantically identical, each entity $e_{i_1j_1}$ is mapped to at most one entity $e_{i_2j_2}$

The central contribution of this paper is to present an approach for defining this mapping function. We only consider one-to-one mappings between single entities. Neither do we cover mappings of whole ontologies or sub-trees, nor complex mappings as concatenation of literals (e.g. name corresponds to first name plus last name) or functional transformation of attributes (e.g. currency conversions).

3 Similarity Measures

Our mapping approach is based on different similarity measures. In this section we want to describe how the various similarity methods have been created.

3.1 Manual Rules

Our implemented approach is based on manually encoded mapping rules. Please note that the mappings itself are not yet encoded through rules (as in [MWK00]). We are using rules to identify possible mappings. This manual effort is necessary because coherences in ontologies are too complex to be directly learned by machines. An expert understanding the encoded knowledge in ontologies formulates machine-interpretable rules out of the information. Each rule shall give a hint on whether two entities are identical, but no rule for itself provides enough support to unambiguously identify a mapping. Naturally, evaluation of these manually created rules has to be a core element of the overall process.

3.2 Similarity Stack

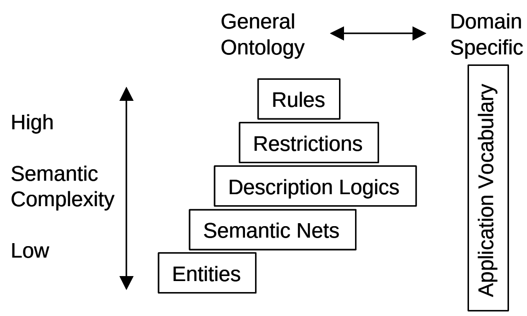

Figure 2: Similarity stack

The presented general idea will now be explicitly used to determine similarity between ontologies. To get a better understanding, the rules are categorized in a similarity stack as shown in graph 2. Ontologies are based on certain vocabularies which are well-defined, well-understood, and with a generally accepted meaning. The left part shows these aspects arranged along their complexity, which is derived from the "layer cake" of [BLHL01]. Special shared ontology domains e.g. SWAP-common in the mentioned SWAP project, have their own additional vocabulary. The right part therefore covers domain-specific aspects. As this domain-specific knowledge can be situated at any level of ontological complexity, it is presented as a box across all of them. In the next paragraphs the general semantic meaning of features is described followed by the concrete derived rules, being tagged with a number (**R**n) with $n \in (1, ..17)$. As many of the rules are derived from existing literature, we give references where applicable.

3.3 Single Similarity Methods

3.3.1 Entities

The first level describes entities as is. No ontological considerations are needed for these features. Labels are human identifiers (names) for entities, normally shared by a community of humans speaking a common language. We can therefore infer that *if labels are the same, the entities are probably also the same* (**R1**, see example 1). Several ideas have already been created to compare labels, e.g. the edit distance[Lev66]. Dictionaries (e.g. WordNet [Fel]) can further be used for comparisons even across languages, although some restrictions apply. Another feature of objects can be an identifier such as URIs, which are unique for every entity. Therefore we know that *if two entities have the same identifier they are identical* (**R2**).

```
<owl:Class rdf:ID=''id1''>
```

```
    <rdfs:label>telephone number</label>
</owl:Class> <owl:Class rdf:ID=''id2''>
    <rdfs:label>phone number</label>
</owl:Class>
```

Example 1. Two entities id1 and id2 with similar labels.

Even without being able to interpret “label” or “URI” it is still possible to do comparisons - based on string or number matching.

3.3.2 Semantic Nets

The second level is the level of Semantic Nets as e.g. introduced by [Qui67]. A concept is a general class of objects. They are in relation to others through attributes or properties. *If the properties of two concepts are equal, the concepts are also equal* (**R3**). The same is true for properties. *If the domain and range (the original and the result concept) of two properties are equal, the properties are also* (**R4**).

3.3.3 Description Logics

The third level described here covers ontologies which have the complexity as provided by Description Logics [BMNPS03]. A taxonomy can be created over concepts, in which a concept inherits all the relations of its super-concepts. Another rule is that if concepts are the same, they will probably have the same super-concepts. We turn the rule around: *if super-concepts are the same, the actual concepts are similar to each other* (**R5**). In practice we calculate the degree of overlap of the two super-concept sets, which provides a number between 0% and 100% [CAFP98]. And finally the sub-concepts of two equal classes will also be the same. *If sub-concepts are the same, the compared concepts are similar* (**R6**) [MMSV02]. Also, *if concepts have similar siblings (i.e. children of parents), they are also similar* (**R7**).

```
<owl:Class rdf:ID=''automobile''/> <owl:Class rdf:ID=''car''/>
<owl:Class rdf:ID=''porsche''>
    <rdfs:subClassOf rdf:resource=''automobile''>
    <rdfs:subClassOf rdf:resource=''car''>
</owl:Class>
<owl:Class rdf:ID=''mercedes''>
    <rdfs:subClassOf rdf:resource=''car''>
    <rdfs:subClassOf rdf:resource=''automobile''>
</owl:Class>
```

Example 2. Two entities with the same sub-concepts.

It is also possible to group properties into a taxonomy, with the corresponding rules resulting: *super-properties* (**R8**) and *sub-properties* (**R9**). The next piece of information which can be added are instances. An instance is a specific entity of a general class from which it inherits all the relations. A concept on the other hand can also be defined as a representative for a set of instances. We can therefore infer that *concepts that have the same instances are the same* (**R10**) [DMDH02]. Vice versa, *instances that have the* *same mother concept are similar* (**R11**). It is also interesting to have a look at the possible distribution of instances on concepts. *If concepts have a similar low/high fraction of the instances, the concepts are similar* (**R12**). Like concepts are interconnected via properties, instances are also regarded to be interconnected via properties instances. This means that *if two instances are linked to another instance via the same property, the two original instances are similar* (**R13**). To a certain degree we can also turn this around: *if two properties connect the same two instances, the properties can be similar* (**R14**).

3.3.4 Restrictions

We continue with ontologies using restrictions. This is covered by e.g. the ontology language OWL. In OWL there are properties such as "sameIndividualAs" or "sameClassAs". *They explicitly state that two entities are the same* (**R15**). A number of further features from OWL could be used, but are discarded at this time, as they do not have any wide distribution yet: property characteristics (symmetric property, functional property, inverse of, inverse functional property), restrictions of values (all values from, some values from, cardinality, has value), equivalence (equivalent class, equivalent property), set operators (same individual as, same as), enumeration (one of), and disjointness (disjoint with). From all of them new rules to determine similarity can be derived.

3.3.5 Rules

Higher levels of the ontology "layer cake" [BLHL01] can also become interesting for similarity considerations. Especially if similar rules between entities exist, these entities will be regarded as similar. For this one would have to process higher-order relationships. Unfortunately there has not been sufficient research and practical support for the rule layer in the Semantic Web in general, not at all for similarity considerations.

3.3.6 Application-specific Vocabulary

All the similarity methods so far use features of the general ontology definition. Besides this one can to have ontologies which an application-specific vocabulary from a certain domain. If the vocabulary is clearly defined and shared it can be taken for additional similarity considerations. As an example we take the ontology used within the SWAP project, in which every file has a unique hash-code assigned. *If the hash-codes of two files are the same, one can infer that they are the same* (**R16**). Additionally, *files with the same MIME-type are similar*, at least in their format (**R17**).

3.3.7 Similarity Paths

In a bigger environment one can expect to have to do more complex mapping e.g. of elements of multiple ontologies. In this case we can use the notion of similarity itself to receive information on other mappings. Similarity as defined here has transitive characteristics if A is similar to B, and B is similar to C, A is similar to C. Some relaxation has to be added when the paths become too long.

	Rule	Name
Concepts are similar, if ... [rule] ... are similar	$R1$	labels
	$R2$	URIs
	$R3$	properties
	$R5$	super-concepts
	$R6$	concept siblings
	$R7$	sub-concepts
	$R10$	instances
	$R11$	fraction of instances
	$R15$	sameAs relation (links the two)
Relations are similar, if ... [rule] ... are similar	$R1$	labels
	$R2$	URIs
	$R4$	domain and range
	$R8$	super-properties
	$R9$	sub-properties
	$R14$	connected instances
	$R15$	sameAs relation (links the two)
Instances are similar, if ... [rule] ... are similar	$R1$	labels
	$R2$	URIs
	$R11$	mother-concepts
	$R13$	properties and instances
	$R15$	sameAs relation (links the two)
Files are similar, if additionally ... [rule] ... are similar	$R16$	hash-code
	$R17$	mime-type

Table 1: Rules and Complexity

In this section we have presented 17 expert encoded rules (Figure 1). For all these similarity measures we now need a way to integrate them.

4 Integration Approach

With the single similarity methods some more steps are required to receive the best possible mappings.

4.1 Combination

According to our hypothesis, a combination of the so far presented rules leads to better mapping results compared to using only one at a time. Clearly not all introduced similarity methods have to be used for each aggregation, especially as some methods have a high correlation. We present both manual and automatic approaches to learn how to combine the methods. Even though quite some methods exist, no research paper focused on the combination and integration of these methods yet. Better mapping results are expected from intelligent approaches to this.

4.1.1 Summarizing

A general formula for this integration task can be given by summarizing over the n weighted similarity methods.

$$sim(e_{i_1 j_1}, e_{i_2 j_2}) = \sum_{k=1}^{n} w_k sim_k(e_{i_1 j_1}, e_{i_2 j_2})$$

with w_k being the weight for a specific method sim_k and $n \in \mathbb{N}$

Please note our assumption that similarities can be aggregated and are increasing strictly. The naive approach would be to simply add the results of the single methods giving them a weight which indicates their importance. The weights could be assigned manually or learned e.g. through maximization of the f-measure (see section 5) of a training set.

The reader might wonder why we don't normalize the sum and take the average to determine equality. In our approach we are basically looking for similarity values supporting the thesis that two entities are equal. If a measure doesn't support the thesis, it still doesn't mean that it's opposing it. These considerations are directly derived from the open world assumption which we respect in this paper.

4.1.2 Sigmoid Function

A more sophisticated approach doesn't only weight the similarity methods but performs a functional computation on each of them. In the given case the most promising function would be the sigmoid function, which has to be shifted to fit our input range of [0 ... 1] (see figure 3).

$$sim(e_{i_1 j_1}, e_{i_2 j_2}) = \sum_{k=1}^{n} w_k \times sig_k(sim_k(e_{i_1 j_1}, e_{i_2 j_2}) - 0.5)$$

with $sig(x) = \frac{1}{1+e^{-ax}}$ and a being a parameter for the slope

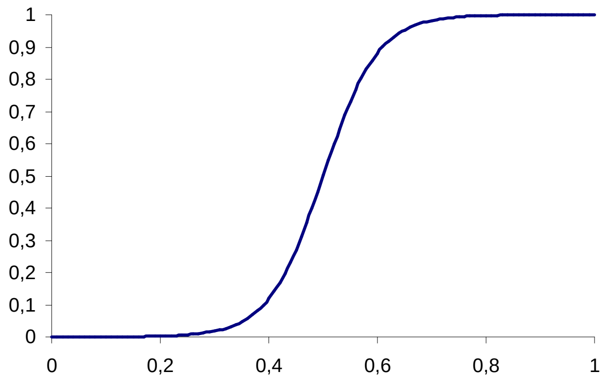

Figure 3: Sigmoid function

The rational behind using a sigmoid function is quite simple: a high similarity value should be weighted over-proportionally whereas a low value practically can be abandoned. An example shall make this clear. When comparing two labels the chance of having the same entity if only one or two letters are different is very high. On the other hand if only three or four letters match there is no information in this similarity at all. The parameters of the sigmoid function can be regarded as an extension of the similarity methods, as they have to be adjusted according to the method they are applied to. A nice side-effect of this intuitive aggregation of methods is that one can easily define the cut-off line for equal entities.

4.1.3 Machine Learning with Neural Networks

A very convenient way of determining how to combine the methods is to use a machine learning approach. As we have continuous inputs, only some machine learning approaches make sense. Support vector machines [Joa98] are one possibility. In our work we focus on neural networks [Hea02]. Specifically specific we chose a three layer neural network consisting of a linear input layer, a hidden layer with a $tanh$ function, and a sigmoid output function (see Graph 4)[7]. A lot of literature discusses how to choose the number of layers, nodes, and edges. We will stick to a simple approach, as we focus on similarity considerations rather than efficient machine learning. The neural network approach also takes the burden from us to determine cut-off values, which is a very important advantage. Unfortunately one needs a big amount of examples to train the network, and always is endangered to overfit the neural network.

4.2 Cut-off

As described earlier, goal of this approach is to find correct mappings between two ontologies. After the just described steps we have a list which consists of the most

[7] http://www.joone.org

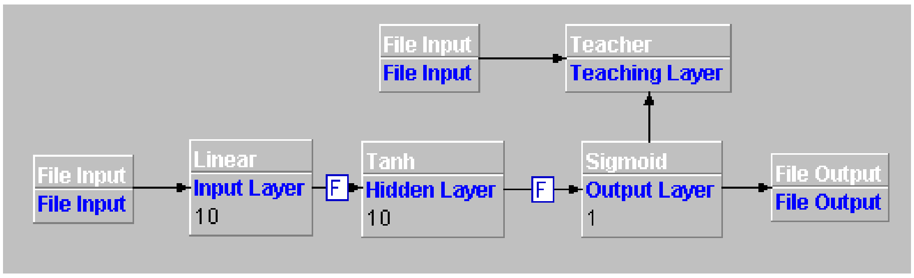

Figure 4: Neural Network (edited with Joone)

similar entities of two ontologies plus the corresponding similarity value. Now remains the question which level of similarity is appropriate to indicate equality for the mapping and which strongly indicates inequality? It is important to make the decision where to put the cut-off. We use the thresholds as cf. [DR02]. Every similarity value above the cut-off indicates a match; everything below the cut-off is dismissed. Three possible heuristics are now described. Still, defining the cut-off is not at all easy or intuitive.

Constant similarity value For this method a fixed constant has to be defined.

$$b = c, \text{ with } b \text{ being the cut-off}$$

All matches with a similarity above this constant are valid. The difficulty is to determine this value. Possible approaches are to take an average which maximizes the f-measure in several test runs. Alternatively it might make sense to use experts to determine the value. Latter only works if the similarity value can be interpreted completely, as one can with the sigmoid summarization.

Delta method In this method the cut-off value for similarity is defined by taking the highest similarity value of all and subtracting a fixed value from it. Again this can be found through maximizing over different f-measures.

$$b = max(sim(e_x, e_y)|\forall e_x \in O_1, e_y \in O_2) - c$$

n percent This method is closely related to the former one. Here we take the highest similarity value and subtract a fixed percentage from it.

$$b = max(sim(e_x, e_y)|\forall e_x \in O_1, e_y \in O_2)(1 - p)$$

The latter two approaches are motivated from the idea that similarity is also dependent on the domain. The calculated maximum can be an indicator for this.

Our approach focuses on classifying the found mappings into two groups: equal or not equal. As a potential extension in future we foresee a *three layer semi-automatic* approach having: correct mappings, mappings to be confirmed manually, and dismissed mappings.

4.3 Additional Actions

Using small additional actions can lead to significantly better results.

4.3.1 Multiple Rounds

For calculating the similarity of one entity pair many of the described methods rely on the similarity input of other entity pairs. The first round always has to be a general method like the comparison based on labels, which does not rely on any other pairs. By doing the calculation in several rounds one can then access the already calculated pairs and receive a better similarity. Several possibilities when to stop the calculations have been described in the literature: a fixed number of rounds, no changes in the mappings, changes below a certain threshold, or dynamically depending on how much time and calculation power can be provided.

4.3.2 Best Mappings Only

When having more than one round of calculation the question arises if the results of each round should be converted/adjusted before they are fed back for the next round. One approach is to reuse only the similarity of the best mappings found. A possible way could be to give the best match a weight of 1, the second best of $\frac{1}{2}$, and the third of $\frac{1}{3}$. Potentially correct mappings are kept with a high probability but leave a path for second best mappings to replace them. The danger of having the system being diverted by low similarity values is minimized.

4.3.3 Deletion of Doubles

The goal of the current approach is to gain a single mapping between two entities from the best similarity values. As there can be only one *best* match, every other match is a potential mistake, which should be dropped. Practically we do cleansing in the mapping table by removing entries with already mapped entities. Consequently the second highest similarity will take over the place of the first one, if the first has been deleted.

The evaluation in Section 5 will show the value of these small but efficient actions.

4.4 Process

All the ideas presented so far describe how two entities can be compared to one another and determine a mapping measure between them. We use the following methodology (see figure 5):

1. Starting point are two ontologies which have to be mapped. We will therefore calculate the similarities between any valid pair of entities.

2. In a first round basic similarities are set via measures which are independent of other similarities. In our case we rely on the label similarity, equal URIs, or the sameAs relation (**R1**, **R2**, and **R15**). The complete similarity matrix is calculated from this. Alternatively a default low similarity value can be chosen.

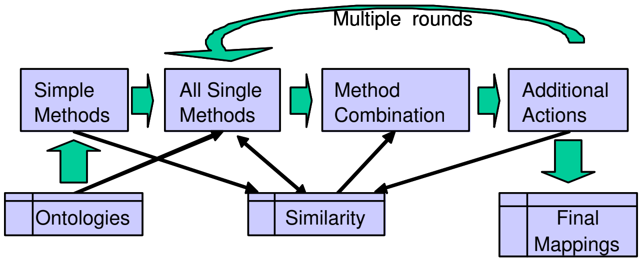

Figure 5: Mapping Process

3. In a second step the overall similarities between the entities are calculated based on all the introduced similarity measures (**R1** through **R17**), always using the now existing previous similarities of other entities if required.

4. Following the presented additional actions steps two and three are repeated for multiple rounds (either a fixed number of times, or until the number of changes per round drops below a threshold value). In a last step doubles are deleted and similarities which are too little (i.e. below the cut-off value and therefore not worth to mention) are removed and only the best similarities are displayed.

5. These will then be used as the final mapping table. They will be evaluated as explained in the next section.

5 Evaluation

The problem of mapping between ontologies already produced some interesting approaches. A thorough evaluation of our new approach is presented here.

5.1 Implementation

From all the presented methods and actions one can derive a strategy to finally determine the similarity between two entities and decide if they can be regarded as equal or not. These will then be tested. The implementation itself was done in Java using the KAON-framework[8] for ontology access and maintenance. All the tests were run on a standard 750 MHz Pentium 3 Notebook.

5.2 Evaluation Scenario

Our evaluation is based on the introductory example given in Section **??**. We have presented an application scenario for mappings at the beginning of this paper.

We basically take two ontologies and create mappings between the entities based on a given strategy. These mappings are validated against the correct mappings which had

[8]http://kaon.semanticweb.org/

been created in beforehand. Our goal was to reach the best number of mappings, which is quantified in the f-measure (see next section). As the absolute quality of mappings is highly dependent of the complexity of the ontologies themselves, we focus on the relative performance of different mapping strategies.

5.3 Metrics

To allow for comparability not only between our own test series, but also with existent literature we will focus on using standard information retrieval metrics. The definitions of precision and recall is adapted by us to fit the given evaluation scenario [DMR02].

Recall describes the number of correct mappings found in comparison to the total number of existing mappings.
$r = \frac{\#correct\_found\_mappings}{\#possible\_existing\_mappings}$

Precision We measure the number of correct mappings found versus the total number of retrieved mappings (correct and wrong).
$p = \frac{\#correct\_found\_mappings}{\#all\_found\_mappings}$

F-Measure is a measure combining the two mentioned measures. It was first introduced by [VR79].
$f = \frac{(b^2+1)pr}{b^2p+r}$ with $b = 1$ being a factor to weight precision and recall.

The given standard measures are used to calculate the following measures from which we expect to gain important insights.

11-Point Measure is the 11-point interpolated average precision at the TREC[9] compititions. We adjusted it to the similarity scenario: eleven points are equally distributed between the best match and the least match. This way we can gain an average precision, recall, or f-measure.

Measures at Cut-Off takes into account how well the algorithm can determine which mappings are still valid and which should be dismissed.

5.4 Data Sets

Four data sets each consisting of at least two ontologies were used for evaluation purposes. From the differences of them we expect a representative evaluation.

Russia 1 In this first set we have two ontologies describing Russia. The ontologies were created by students with the task to represent the content of two independent travel websites about Russia. These ontologies have approximately 400 entities, including concepts, relations, and instances. The total number of theoretical mappings is at 280, which have been assigned manually. This scenario is an easy scenario, with which many individual methods can be tested.

[9] http://trec.nist.gov/

Russia 2 The second set again covers Russia. This time the two ontologies have been additionally altered by deleting entities and changing the structure as well as the labels at random. Each ontology has about 300 entities with 215 possible mappings, which were captured during the generation. Many of these mappings can not even be identified by humans any longer.

Tourism Two ontologies which were created separately by different groups of people describe the tourism domain of Mecklenburg-Vorpommern (a federal state in the north-east of Germany). Both ontologies consist each of about 500 concepts and relations, but no instances though. 300 manual mappings were created. The interesting part of this scenario is the question how missing information (in our case the missing instances) affects the outcomes of our algorithm.

SWRC The SWRC (Semantic Web Research Community) ontology describes the domain of universities and research. Its size is about 300 entities, with very little instances. For this setting three more very small ontologies (about 20 entities each) were created. In total we have 20 possible mappings (manually determined) against the SWRC ontology. This scenario was derived from the SWAP case where small queries (plus additional context) are sent to an existing ontology for validation.

From the differences of the data sets we expect a representative evaluation.

5.5 Strategies

For the tests we chose to use five similarity strategies:

Label (S1) For this strategy only the labels of entities were regarded (**R1**). This strategy can be regarded as the baseline against which we need to evaluate the other strategies with more advanced measures.

All (S2) As a next step all described similarity methods (**R1** through **R15**) are integrated through simple addition.

Weighted (S3) All similarity methods are integrated including different weights for each method. The weights were calculated by maximizing the overall f-measure in the four test data sets. Additionally five rounds of similarity calculation are done and doubles are removed.

Sigmoid (S4) Again all methods (**R1** to **R15**) are taken, but they are weighted with the sigmoid function. The parameters of the sigmoid functions were assigned manually with respect to the underlying similarity method. In the five rounds only the best results were fed back into the next round. Finally doubles were removed. A constant was used to determine the cut-off.

Neural Nets (S5) The results of the methods are fed into a neural network. A fraction (20%) of the evaluation examples was taken for training purposes. The rest was then used for evaluation. A constant value for cut-off was determined from the same training set manually.

5.6 Results

For space purposes we will only present the results of the first data set graphically (Figures 6 through 10)[10]. The dark curve represents the calculated similarity. All matches are arranged by their similarity value (see Table 2) - with the left side of the graph showing the highest value and the right side showing the lowest value. With each new match we recalculate the other measures. Each graph shows the respective precision, recall, and f-measure value. The vertical line shows the cut-off border. This is the point we have to measure if we are looking for exactly one value. Alternatively we also measured the 11-point average to gain a view of the whole curve. The left axis describes the value of recall, precision, and f-measure; the right axis describes the value of the similarity methods.

Entity 1	Entity 2	Similarity
http://lone.org/russia#music	meh://8807#music	1
http://lone.org/russia#haseconomofact	meh://8807#haseconomicalfact	0,9473
http://lone.org/russia#costmoneyeating	meh://8807#costmoney	0,9473
http://lone.org/russia#includecity	meh://8807#includetown	0,9167
...	...	...

Table 2: Excerpt of the Similarity Result Table

In the beginning all strategies have correct mappings, but the precision gets worse very fast with some of them. Only regarding labels already has very good results (Figure 6). By just combining semantical measures blindly we don't gain anything, except for a slightly higher overall recall level (Figure 7). But, as soon as we use some more sophisticated methods and actions the results become better (Figure 8). Precision stays high much longer than by just using labels. The overall recall also achieves higher levels resulting in an overall higher f-measure. The advanced sigmoid doesn't gain too much when looking at the curves (Figure 9), but allows a much clearer cut-off. Whereas the cut-off of other strategies seems to be chosen randomly (it was actually tried to maximize the f-measure over all tests), the fixed value of $c = 2$ [11] almost hits the maximum f-measure in every test. Finally we used an approach were the weights were learned automatically through a neural network. The results were less promising than we had expected (Figure 10). It seems that precision almost reaches the manual sigmoid level, but recall is lower than with any other method.

The three other data set results are summarized in Table 3. Here we only wrote down the 11-point average and the measures using a fixed constant c for cut-off or a value based on subtracting a fixed amount delta.

[10]We refer to http://www.aifb.uni-karlsruhe.de/WBS/meh/publications/ESWS for the complete graphs as well as the used ontologies.

[11]The cut-off $c = 2$ was chosen, because this represents the standard value of 1 increased by a value of 1 for full similarity support.

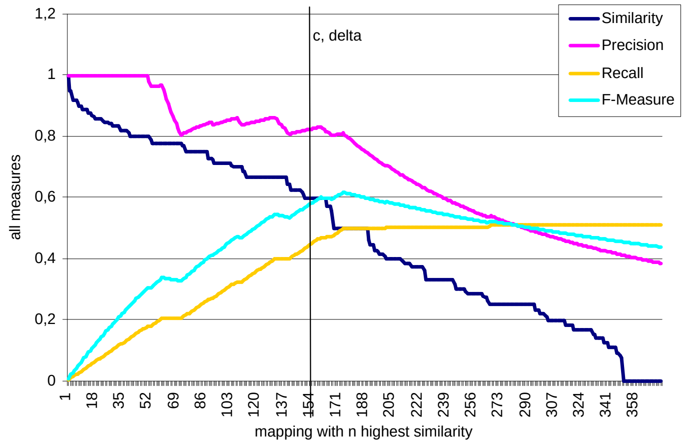

Figure 6: Russia 1, Labels

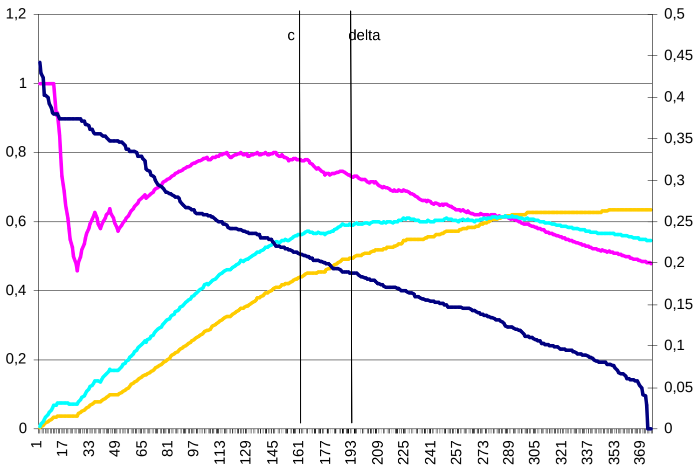

Figure 7: Russia 1, All

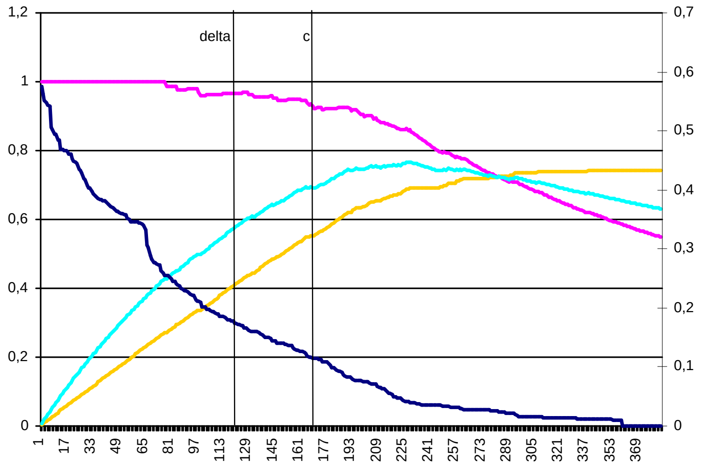

Figure 8: Russia 1, Linear

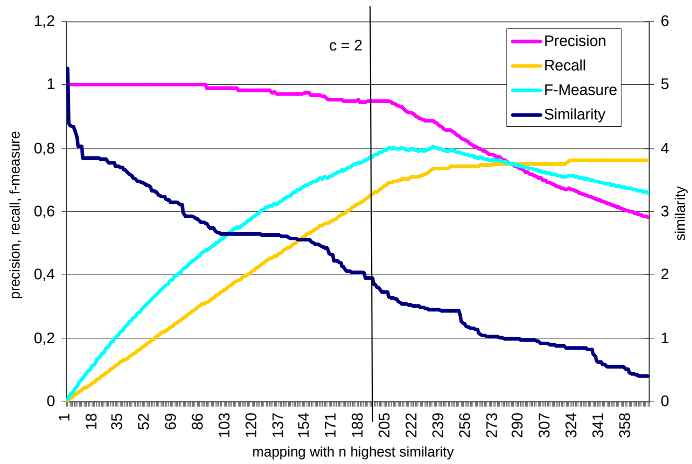

Figure 9: Russia 1, Sigmoid

	duration in sec.	11-point precision	11-point recall	11-point f-measure	precision c	recall c	f-measure c	precision delta	recall delta	f-measure delta
russia1 label	30	0,7012	0,3768	0,428	0,8023	0,4734	0,5955	0,8023	0,4734	0,5955
russia1 all	133	0,6856	0,4086	0,4516	0,7459	0,4876	0,5897	0,7135	0,5194	0,6012
russia1 linear	525	0,8509	0,4998	0,5575	0,9209	0,5759	0,7086	0,9607	0,3462	0,5090
russia1 sigmoid	520	0,8692	0,5146	0,5735	0,9473	0,6360	0,7610			
russia1 neural net	1103	0,7887	0,4391	0,4977	0,9718	0,4876	0,6494			
russia2 label	20	0,3525	0,2179	0,2475	0,5346	0,2583	0,3483	0,5346	0,2583	0,3483
russia2 all	116	0,4914	0,2414	0,2746	0,4500	0,2583	0,3282	0,3986	0,2822	0,3305
russia2 advanced linear	520	0,5065	0,2644	0,2978	0,4758	0,3301	0,3898	0,5277	0,1818	0,2704
russia2 advanced sigmoid	411	0,6699	0,3632	0,4111	0,7192	0,3923	0,5077			
russia1 neural net	875	0,5994	0,3062	0,350496814	0,7763	0,2822	0,4140			
tourism label	51	0,7733	0,6540	0,6119	0,8206	0,8472	0,8337	0,8206	0,8472	0,8337
tourism all	986	0,7686	0,6531	0,6075	0,5217	0,9444	0,6721	0,5929	0,9305	0,7243
tourism advanced linear	5132	0,7844	0,6632	0,6214	0,6945	0,8842	0,7780	0,6518	0,8842	0,7504
tourism advanced sigmoid	4155	0,7809	0,6620	0,6188	0,8851	0,8564	0,8705			
russia1 neural net	9146	0,7760	0,6489	0,6097	0,9471	0,7454	0,8342			
swrc label	12	0,6888	0,5505	0,5325	0,7647	0,7222	0,7428	0,7647	0,7222	0,74285
swrc all	18	0,4143	0,4090	0,3800	0,5	0,2778	0,3571	0,5625	0,5	0,5294
swrc advanced linear	40	0,4292	0,4646	0,4261	0,6667	0,6486	0,5882	0,5454	0,6667	0,6
swrc advanced sigmoid	40	0,8058	0,6667	0,6417	1	0,7222	0,8387			
russia1 neural net	89	0,6942	0,5303	0,5223	0,75	0,6667	0,7059			

Table 3: Test results

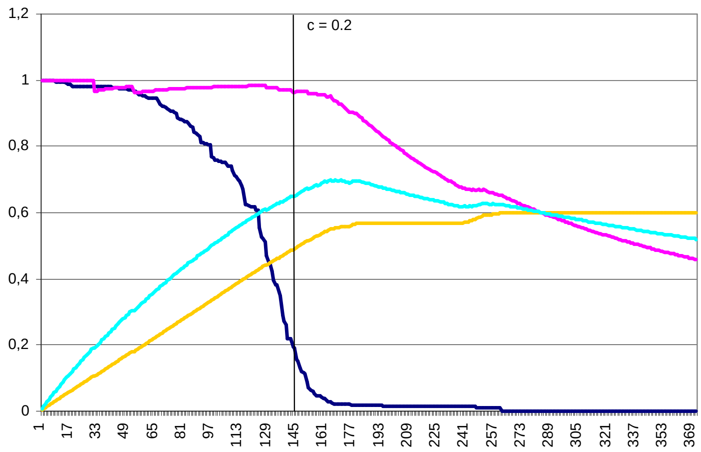

Figure 10: Russia 1, Neural Net Trained

5.7 Discussion

As a hypothesis we expected better mapping results from intelligent approaches in combining single similarity methods. A general comparison graph (figure 11) is plotted at the end of this section. This comparison graph shows the average results over all four tests. We will now discuss these results in more detail.

As a baseline for comparison we used strategy S1 “Label”.

One can see very well that the combination of methods leads to a significant increase of recall, precision and f-measure. The semantical rich structures do help to determine similarity. It also seems that by simply adding the results of the methods linearly results are not always better. Only small gains are found for some measures. It is especially difficult to determine the cut-off point, which we require for automatic mapping. Using the sigmoid function for weighting finally does provide the full power of semantical rich structures. For every single plotted measure the results are better than by just using labels. In our evaluation the values of precision, recall and f-measure show an average increase of 20%.

Machine Learning might help. The problems we encountered might be a general problem of machine learning. The training algorithm tries to gain most for the given examples. Unfortunately this inherently leads to over-fitting to a certain degree. When ontologies diversing structurally from the training ontologies are supposed to be mapped, results drop drastically. Vice versa manually assigned weights are more abstracted from the given ontologies and therefore perform better. A second problem was to model the additional actions e.g. the modelling of removing doubles could not be fed into the neural network resulting in a network not optimized on the whole integrated mapping process.

A short comment about the absolute quality of mappings has to be made at this point. Some mappings are simply not identifiable, not even by humans. This is the

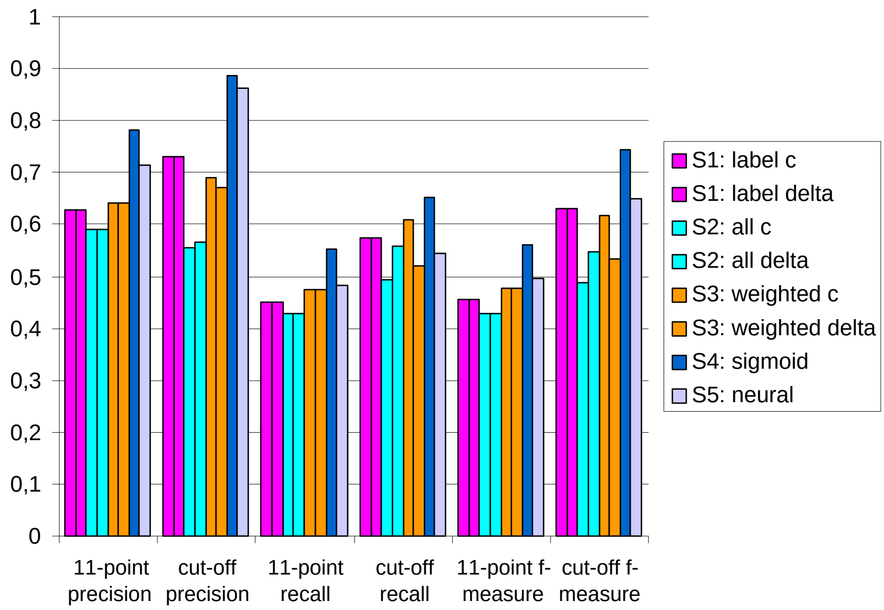

Figure 11: Graphical Results, averaged over tests

reason why some results only reach an unsatisfying level of recall. As a result the above considerations were based on relative performance of different measures among each other.

5.8 Implications

As a hypothesis we expected better mapping results from intelligent approaches in combining single similarity methods. This was the idea we had when developing the new combination methods. Our original hypothesis H is widely fulfilled:

Semantics can help to determine better mappings (S1 vs. S4).

Precision is considerably higher for the more advanced combination methods. Especially interesting is the fact that precision generally is higher for these methods, no matter where the cut-off is placed in the mapping table. This is important when thinking of full-automatic solutions, where we want to keep the wrong mappings as low as possible.

Recall also rises along the richness of methods.

F-measure as our core evaluation measure reaches the highest value for the S4 Sigmoid strategy for every data set.

Average Increase of 20% in precision, recall and f-measure.

Naive Combinations of mapping methods often do not make the results better, but worse (S2 All, and even S3 Weighted) . The effort for advanced carefully determined methods is therefore very important.

Machine Learning might help (S5). The problems we encountered are general problems of machine learning such as over-fitting. We also faced the problem that the additional actions could not be completely integrated into the machine learning approach, which lead to lower results.

The results are very promising and motivate to continue work on this.

6 Outlook

6.1 Related Work

Most of the ideas for measuring similarity are derived from common sense and can be easily understood. To our knowledge the existing approaches focus on specific methods to determine similarity rather than using an overall approach.

Some authors have tried to find a general description of similarity with several of them being based on knowledge networks. [RE00] give a general overview of similarity. [RE99] try to include different kinds of context. Already some years old and not directly addressing ontologies but rather knowledge structures is the work [Fra93]. However, all these approaches often lack the specific application for ontologies and significant testing.

As the basic ontology mapping problem has been around for some years first tools have already been developed to address this. The tools PROMPT [NM02] and Anchor-PROMPT [NM01] use labels and to a certain extent the structure of ontologies. Their focus lies on ontology merging i.e. how to create one ontology out of two. The quality of the automatic methods are checked by a human by verifying the actions before they are processed. Anchor-PROMPT can therefore be categorized to the semi-automatic tools. [DMDH02] already used a general approach of relaxation labelling in their tool GLUE. Most of their work is based on the similarity of instances though. It might be sufficient enough in environments with many instances, but will create problems in other domains, if less instances are existent. [McG00] created a tool for mapping called Chimaera. Potential matches are presented to the user including the hierarchical order. This allows an easier decision for the user.
Besides equality first steps are taken in the direction of complex matches. These could also include concatenation of two fields such as “first name” and “last name” to “name”[DR02]. [BMSZ03] further present an approach for semantic mappings based on SAT. This is not regarded in this paper, but may show the direction of future work.

Despite the large number of related work, there are very little approaches on how to combine the many methods. [DMDH02] give a first hint, but do not continue the research on it. The above mentioned tools (PROMPT, GLUE, etc.) also require some kind of integration, but do not raise the issue. The presumably use only naive summarization approaches. Solving the integration aspects was part of the motivation of this paper.

[RHdV03] express their insights from a database view. Many ideas from the database community, especially concerning efficiency (see [YMK02] and [MNU00]), should also be regarded when talking about ontologies. Another community involved in similarity and mapping are object-oriented representations [BS98]. [Ruf03] did this for UML. When having agents communicating mapping becomes important as shown in [WB99]. This is a little related to the described peer-to-peer scenario this work was originally based on.

6.2 Problems and Future Steps

Even though the shown approach retrieves good results, the results are not 100% correct. This might be tolerable in some scenarios. Unfortunately, if full-automatic mapping is done, and inferencing builds on top of it, wrong results can bring down the value of the whole mapping process. Implications of this will have to be understood well when using it. A common approach to circumvent this problem is to declare the process as semi-automatic rather than doing full-automatic mapping. Nevertheless there is a general problem on how to deal with inexact or fuzzy inferencing and fuzzy results in the ontology domain. To our knowledge this area hasn't been thoroughly investigated yet, but is started [DP04].

Another problem is a general problem when doing comparisons. Especially with big ontologies complexity of similarity calculations can grow exponentially. Already by comparing every entity with every other entity the approach becomes infeasible for big amounts of data. By adding complex similarity measures things get even worse. One can expect a complexity of $O(log(n)^2 \times n^2)$. It is derived from: $O(log(n))$ for entity access, $O(1)$ for the method complexity, and $O(n^2)$ for the full comparison of all possible pairs. Approaches from other domains (e.g. databases [MNU00]) to reduce complexity might be a good start. As data in ontologies expresses certain semantics the calculations might be channelled using these semantics e.g. starting with comparisons of top-level elements in the hierarchy. This is another direction of our future work.

6.3 Conclusion

The mapping problem arises in many scenarios. We have shown a methodology for identifying mappings between two ontologies based on the intelligent combination of manually encoded rules. Evaluation proved our initial hypothesis, i.e. the combination of our presented similarity measures leaded to considerably better results than the usage of one at a time. One can summarize that precision, recall, and f-measure increase by 20% compared to label-based approaches. Semantics helps bridging the mapping gap.

Acknowledgements. Research reported in this paper has been partially financed by EU in the IST project SWAP (IST-2001-34103). Many thanks to our colleagues for the fruitful discussions.

References

[Bis95] G. Bisson. Why and how to define a similarity measure for object based representation systems. *Towards Very Large Knowledge Bases*, pages 236–246, 1995.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[BMNPS03] Franz Baader, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook*. 2003.

[BMSZ03] Paolo Bouquet, B. Magnini, L. Serafini, and S. Zanobini. A SAT-based algorithm for context matching. In *IV International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'2003)*, Stanford University (CA, USA), June 2003.

[BS98] Ralph Bergmann and Armin Stahl. Similarity measures for object-oriented case representations. *Lecture Notes in Computer Science*, 1488:25+, 1998.

[CAFP98] S. V. Castano, M. G. De Antonellis, B. Fugini, and C. Pernici. Schema analysis: Techniques and applications. *ACM Trans. Systems*, 23(3):286–333, 1998.

[CGL$^+$04] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. What to ask to a peer: Ontology-based query reformulation. In *Proc. of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2004)*, 2004.

[DFKO01] Ying Ding, Dieter Fensel, M. Klein, and Boris Omelayenko. Ontology management: Survey, requirements and directions. 2001.

[DMDH02] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to map between ontologies on the semantic web. In *Proceedings to the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, USA, May 2002.

[DMR02] H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proceedings of the second int. workshop on Web Databases (German Informatics Society)*, 2002.

[DP04] Zhongli Ding and Yun Peng. A probabilistic extension to ontology language owl. In *Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37)*, Big Island, Hawaii, January 2004.

[DR02] H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[Fel] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*.

[Fra93] Robert K. France. Weights and measures: An axiomatic model for similarity conputations. Technical report, Virginia Polytechnic Institute and State University, 1993.

[Gru93] Tom R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[Hea02] Jeff Heaton. *Programming Neural Networks in Java*. 2002.

[Joa98] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[Lev66] I. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 1966.

[McG00] Deborah L. McGuinness. Conceptual modeling for distributed ontology environments. In *International Conference on Conceptual Structures*, pages 100–112, 2000.

[MMSV02] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. Mafra - a mapping framework for distributed ontologies. In *Proceedings of the EKAW 2002*, 2002.

[MNU00] Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining*, pages 169–178, 2000.

[MW] Merriam-Webster. Dictionary.

[MW01] Prasenjit Mitra and Gio Wiederhold. An ontology-composition algebra. Technical report, Stanford University, Stanford, California, USA, 2001.

[MWK00] Prasenjit Mitra, Gio Wiederhold, and Martin Kersten. A graph-oriented model for articulation of ontology interdependencies. *Lecture Notes in Computer Science*, 1777:86+, 2000.

[NM01] Natalya F. Noy and Mark A. Musen. Anchor-prompt: Using non-local context for semantic matching. In *Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, WA, 2001.

[NM02] Natalya F. Noy and Mark A. Musen. The prompt suite: Interactive tools for ontology merging and mapping. Technical report, Stanford Medical Informatics, Stanford University, Stanford, California, USA, 2002.

[Qui67] M. Ross Quillan. Word concepts: A theory and simulation of some basic capabilities. *Behavioral Science*, 12:410–430, 1967.

[RE99] M. Andrea Rodrguez and Max J. Egenhofer. Putting similarity assessments into context: Matching functions with the user's intended operations, 1999.

[RE00] M. Andrea Rodrguez and Max J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 2000.

[RHdV03] John Roddick, Kathleen Hornsby, and Denise de Vries. A unifying semantic distance model for determining the similarity of attribute values. In *Proceedings of the 26th Australsian Computer Science Conference (ACSC2003)*, Adelaide, Australia, 2003.

[Ruf03] Raimi Ayinde Rufai. Similarity metric for uml models. Master's thesis, King Fahd University of Petroleum and Minerals, 2003.

[SEH+03] Gerd Stumme, Marc Ehrig, Siegrfried Handschuh, Andreas Hotho, Alecander Maedche, Boris Motik, Daniel Oberle, Christoph Schmitz, Steffen staab, Ljiljana Stojanovic, Nenad Stojanovic, Rudi Studer, York Sure, Raphael Volz, and Valentin Zacharias. The Karlsruhe view on ontologies. Technical report, University of Karlsruhe, Institute AIFB, 2003.

[Sow97] John Sowa. Electronic communication in the onto-std mailing list, 1997.

[Su02] Xiaomeng Su. A text categorization perspective for ontology mapping. Technical report, Department of Computer and Information Science, Norwegian University of Science and Technology, Norway, 2002.

[VR79] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[WB99] Peter C. Weinstein and William P. Birmingham. Agent communication with differentiated ontologies: eight new measures of description campatibility. Technical Report CSE-TR-383-99, 7, 1999.

[YMK02] Liu Yaolin, Martin Molenaar, and Menno-Jan Kraak. Semantic similarity evaluation model in categorical database generalization. In *Symposium on Geospatial Theory*, 2002.