Instance-based matching of hierarchical ontologies

Andreas Thor, Toralf Kirsten, Erhard Rahm
University of Leipzig
{thor,tkirsten,rahm}@informatik.uni-leipzig.de

Abstract: We study an instance-based approach for matching hierarchical ontologies, such as product catalogs. The motivation for utilizing instances is that metadata-based match approaches often suffer from semantic heterogeneity, e.g. ambiguous concept names, different concept granularities or incomparable categorizations. Our instance-based match approach matches categories based on the instances (e.g. products) assigned to them. This way we partly translate the ontology match problem into an instance match problem which is often easier to solve, especially when instances carry globally unique object ids. Since concepts of different ontologies rarely match 1:1 we propose to determine correspondences between sets of concepts. We experimentally evaluate the match approaches for real product catalogs.

1 Introduction

Ontologies become increasingly important in both commercial and scientific application domains. Relevant objects of such domains, e.g. products, genes, etc., can be semantically described and categorized by ontologies. Typically, such ontologies use a controlled vocabulary for the naming of concepts. Concepts can be organized within several generalization/specialization hierarchies (is-a relationships) and be interconnected by additional relationships. Some ontologies, e.g. in life sciences, aim at providing a shared and standardized description of concepts of a community to help exchange and integrate data from different sources [DH05, WVV+01].

Unfortunately, ontologies also introduce semantic heterogeneity since many independently developed ontologies are now in common use. This is especially the case for organization-specific ontologies such as product catalogs, which are typically designed for a specific purpose. Hence ontologies of different organizations may widely differ even if they address the same application domain. As an example, Figure 1 shows portions of two product ontologies of the e-shops Amazon[1] (left side) and Softunity[2] (right side). Users can browse through the concepts (categories) of such product catalogs to find the associated products, e.g. software products such as "*Windows XP Home*" and "*SuSE Linux 10.1*". Product information is typically structured according to a database schema using product- and shop-specific attributes, such as id, title and price. As the example shows, both ontologies are differently organized. Unlike Softunity, the Amazon ontology consists of multiple orthogonal hierarchies, e.g. "*by brands*" and "*by category*". Therefore, products such as "*Windows XP Home*" can be related with multiple concepts.

[1] http://www.amazon.com
[2] http://www.softunity.com

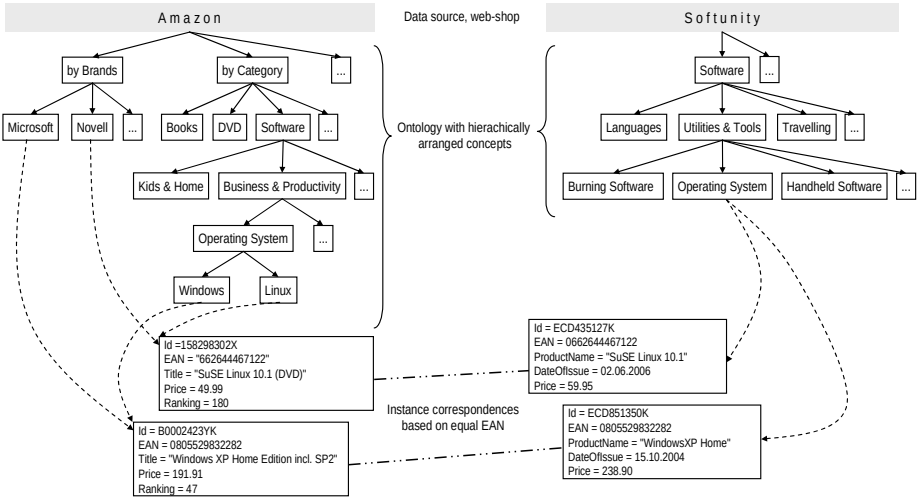


Figure 1: Portions of two application-specific ontologies with associated objects

Moreover, the Amazon ontology differentiates between "*Windows*" and "*Linux*" operating systems while the Softunity ontology only has a single concept "*Operating System*". Hence, both ontologies are of different granularity.

An ontology mapping can bridge the semantic heterogeneity of different ontologies and thus help to search or query data from different sources, e.g. to compare or recommend similar products offered in different e-shops. Previous approaches to determine a mapping or match result between ontologies mostly utilize metadata like the concept names, concept descriptions or structural context information. However, the usefulness of such approaches is often limited due to the semantic heterogeneity problems discussed, e.g. ambiguous concept names, different concept granularities or incomparable categorizations.

We therefore advocate for a simple instance-based match approach which matches concepts (product categories) based on the instances (e.g. products) assigned to them. This is motivated by the assumption that the real semantics of a concept is often better defined by the actual instances assigned to the concept but by metadata like the concept name. To determine matching concepts using instances we need to find matching instances between the ontologies, i.e. we partly turn the ontology match problem into an instance (object) match problem. Instance matching is based on specific data values and thus often easier to solve than matching abstract metadata. An ideal case for instance matching is given when instances carry globally unique object ids. For example, many e-shops use unique product ids, so-called EANs (European Article Number). In the example in Fig.1, the EAN values allow us to find the two shown instance (product) correspondences for the Linux and XP products. These instance correspondences in turn can be used to determine matches between the associated product categories, e.g. we can find out that the Amazon categories “Microsoft” and “Windows” both match the Softunity category “Operating System”. Obviously such an instance-based match approach is the more promising the higher the instance overlap of the ontologies.

Previous match approaches often restrict themselves to mappings of 1:1 and N:1 cardinality. For schema matching such mappings are needed for data exchange between a source and a target schema where each target attribute value must be uniquely derived from one or several source attribute values. Ontology mappings of cardinalities 1:1 and N:1 are sufficient to express equivalence and subset relationships between concepts of different ontologies. However, we find that concepts like product categories of different ontologies may overlap in almost arbitrary ways so that there is a need to support N:M match relationships. We thus propose to use instance matches for determining correspondences between sets of concepts and support 1:1, N:1 and N:M mapping cardinalities. The coarser N:M ontology mappings are still useful for important applications, e.g. ranked keyword queries or product recommendations from related categories at a different e-shop.

The rest of this paper is organized as follows. In the next section we briefly discuss some additional related work. Section 3 describes how to determine instance-based ontology mappings and presents an experimental comparison of its effectiveness with a name-based match scheme. In Section 4 we illustrate and evaluate set correspondences. Section 5 concludes.

2 Related Work

There is a big literature on algorithms for schema matching and ontology matching [RB01, KS03, AGY05, DH05, SE05]. The approaches can be roughly classified as metadata-based, instance-based or mixed forms. Metadata-based match algorithms, e.g. [MS02, ELT+03, NM03, MB04, ADMR05], utilize concept names, concept descriptions or definitions (if available) and the ontology graph structure. However, concept names in e-Business are often short and ambiguous. For instance, concept names, such as "*miscellaneous*", "*collections*" and "*accessories*", are often used in different contexts within the ontology. To make concept names more meaningful, they can be concatenated along the path from the ontology root to the concept node. However, using such path names is not always effective since concepts can be differently arranged in different ontologies by incomparable classification criteria.

Some instanced-based schema matching approaches utilize previously identified duplicate instances between overlapping sources, e.g. [PE95, CCL03, BN05]. While we use instance matches to derive category matches these approaches focus on the use of duplicates for matching the attributes of the instances. Moreover, these approaches consider 1:1 and 1:N/N:1 match cardinalities whereas our ontology matching approach also detects N:M match relationships.

Instance-based ontology matching is investigated in [AS01, ITH03, DMD+03, HYN+04] using different statistical or machine learning approaches. [AS01, DMD+03] utilize a Naïve Bayes classification approach to assign source concepts to the concepts of a master catalog; the instance mapping is used to improve the classification accuracy. [ITH03] matches categories between two internet directories based on their containing web links (instances) but apply a metric that is different from ours. [HYN+04] compares feature vectors for each concept pair using keywords found in the instances and then determines similar feature vectors by a structural matcher. The ontology mappings gen-

erated by all these instance-based approaches only consist of single concept correspondences but not set correspondences.

The evaluation of match algorithms typically requires generated mappings to be compared with a perfect, manually determined match result by using information retrieval metrics such as precision and recall. However, creating such a perfect mapping for large real-world ontologies is extremely labor-intensive. Furthermore, it is often difficult to clearly decide when two concepts should match due to the mentioned problems of semantic heterogeneity. Therefore, we do not try to derive a perfect mapping for our evaluation but compare the result sets of different algorithms with each other, similar to [BAB05, MTM+06].

3 Instance-based Matching of Ontologies

For our study, an ontology consists of a *is-a* hierarchy of concepts. Concepts can have multiple associated instances, i.e., objects that are described or classified by the concept. An instance can be associated with multiple concepts, e.g. when the ontology contains concepts of orthogonal aspects. Moreover, an instance may be assigned not only to leaf-level concepts but also to inner concepts of the ontology.

The key idea of our approach is to derive the similarity between concepts from the similarity of the associated instances. Determining such instance matches is easy in some domains, e.g. by using the non-ambiguous EAN in e-commerce scenarios. Moreover, instance matches may be provided by hyperlinks between different data sources and, thus, can easily be extracted. In the absence of unique identifiers, instance matching can be performed by general object matching (duplicate identification) approaches, e.g. by comparing attribute values.

An important advantage for instance-based ontology matching is that the number of instances is typically higher than the number of concepts. This way, we can determine the degree of concept similarity based on the number of matching instances. Furthermore, the match accuracy of the approach can become rather robust against some instance mismatches.

In the following we first introduce three metrics to determine an instance-based similarity between concepts. Afterwards we present the metrics used for evaluating the ontology match approaches. Section 3.3 evaluates the approaches for matching two real-world product catalogs.

3.1 Similarity metrics

In this paper we study three metrics for determining the *instance-based similarity* between concepts c_1 and c_2 of different ontologies, namely the dice similarity $Sim_{DICE}(c_1,c_2)$, the minimum similarity metric $Sim_{MIN}(c_1,c_2)$ and the base similarity metric $Sim_{Base}(c_1,c_2)$.

The dice similarity metric [Rijs79] between two concepts c_1 and c_2 of the concept sets C_{O1} and C_{O2} of two ontologies O_1 and O_2 is defined as follows:

$$Sim_{DICE}(c_1,c_2) = \frac{2 \cdot |I_{c_1} \cap I_{c_2}|}{|I_{c_1}| + |I_{c_2}|} \in [0...1], \forall c_1 \in C_{O1}, c_2 \in C_{O2}$$

In the formula, $|I_{c1}|$ ($|I_{c2}|$) denotes the number of instances that are associated to the concepts c_1 (c_2). $|I_{c1} \cap I_{c2}|$ is the number of matched instances that are associated to both concepts, c_1 and c_2. In other words: the similarity between concepts is the relative overlap of the associated instances.

The dice similarity values do not take into account the relative concept cardinalities of the two ontologies but determine the overlap with respect to the combined cardinalities. In the case of larger cardinality differences the resulting similarity values thus can become quite small, even if all instances of the smaller concept match to another concept. We therefore additionally utilize the minimal similarity metric which determines the instance overlap with respect to the smaller-sized concept:

$$Sim_{MIN}(c_1, c_2) = \frac{|I_{c_1} \cap I_{c_2}|}{\min(|I_{c_1}|, |I_{c_2}|)} \in [0...1], \forall c_1 \in C_{O1}, c_2 \in C_{O2}$$

For comparison purposes we also consider a *base similarity* which matches two concepts already if they share at least one instance.

$$Sim_{Base}(c_1, c_2) = \begin{cases} 1 \ , & if \ |I_{c_1} \cap I_{c_2}| > 0 \\ 0 \ , & if \ |I_{c_1} \cap I_{c_2}| = 0 \end{cases} \in [0...1], \forall c_1 \in C_{O1}, c_2 \in C_{O2}$$

Obviously it holds for all correspondences between concepts c_1 and c_2:

$$Sim_{DICE}(c_1, c_2) \leq Sim_{MIN}(c_1, c_2) \leq Sim_{Base}(c_1, c_2) \cdot$$

We may also apply other similarity metrics, e.g. an asymmetrical metric such as $Sim(c_1,c_2) = |I_{c1} \cap I_{c2}| / |I_{c1}|$. We leave the analysis of other metrics as a subject for future work.

3.2 Evaluation metrics

The standard metrics for evaluating the effectiveness of match approaches, recall and precision, require that the perfect match result is known. However, this perfect match result is generally unknown for difficult real-life match problems, especially for large heterogeneous ontologies. Fortunately, for our instance-based match approaches we can use the base similarity metric as a yardstick for evaluating alternate match approaches. This is because a baseline matcher using this similarity metric achieves the maximal possible recall for instance-based ontology matching. On the other hand, its precision is likely to be very low because it matches two concepts already if they share only one instance, i.e., even for low concept similarity. Other instance-based approaches (like using the dice or minimum similarity metrics) yield subsets in both the set of matching categories and the correspondences, i.e. lower recall, than the baseline matcher. However, these alternatives are likely to be more precise than the baseline matcher since they restrict themselves to category correspondences with a larger instance overlap.

For measuring the recall of a match approach we thus propose to use a relative *Match-Coverage* metric w.r.t. to the baseline matcher. Let $Corr_{O1\text{-}O2}$ be the number of determined correspondences between ontologies O1 and O2 for a given match approach. C_{O1} (C_{O2}) denotes the set of matched O1 (O2) concepts, i.e., the set of concepts having at least one correspondence. We then define match coverage as follows:

$$MatchCoverage = \frac{|C_{O1}| + |C_{O2}|}{|C_{Base-O1}| + |C_{Base-O2}|}$$

Table 1: Quantity structure of concepts and associated instances

	Softunity	Amazon
# Concepts (product categories)	470	1,856
# Concepts having directly associated instances	170	1,723
# Instances (products)	2,576	18,024
# Direct associations	2,576	25,448
# Direct associations / # Instances	1	≈ 1.4
# Instances / #concepts (directly associated)	≈15	≈15

In the formula, $C_{Base\text{-}O1}$ ($C_{Base\text{-}O2}$) is the set of matched O1 (O2) concepts using the baseline approach.

For estimating the precision of a match approach we determine the so-called *MatchRatio* metric, i.e., the ratio between the number of found correspondences and the number of matched concepts:

$$MatchRatio_{O1} = \frac{|Corr_{O1-O2}|}{|C_{O1}|} \qquad MatchRatio_{O2} = \frac{|Corr_{O1-O2}|}{|C_{O2}|}$$

The intuition is that the value (precision) of a match result is better if a concept is not loosely matched to many other concepts but only to fewer (preferably the most similar) ones. The match ratio for the baseline matcher is expected to provide a worst-case value for instance-based matching.

3.3 E-Commerce scenario

Our experimental evaluation uses the real-world product catalogs and instance data of Amazon.de and Softunity.com. The catalogs are restricted to the area of software and games. Table 1 summarizes their characteristics. The comparison of Amazon and Softunity shows a significant difference in both the number of instances and the number of concepts. Note that, unlike Softunity, Amazon products are on average directly associated to 1.4 concepts. Only 36% of all Softunity concepts have directly associated products but almost 93% of Amazon concepts do so. Obviously, Amazon frequently associates products to inner concepts that are less related with their descendants in the hierarchy. Note that concepts also have *indirectly associated products*, i.e. the products which are directly assigned to at least one of their descendants.

The underlying (perfect) instance match is determined by matching products having the same EAN. It contains 1872 matches and cover about 73% of the Softunity products. Using the perfect instance mapping we determine correspondences based on the introduced similarity metrics. Table 2 shows the results for the baseline matcher; Table 3 and Fig. 2 show results for the Dice and Minimum similarity metrics for different similarity thresholds. In all cases, we distinguish between direct associations (concept similarity based on overlap of directly associated instances), and indirect associations that also consider instance associations from sub-concepts of the is-a hierarchy. For indirect associations we eliminate trivial concept correspondences, i.e., given a correspondence between two concepts we remove all correspondences between their ancestors that do not have a greater similarity. For a given threshold, the usage of indirect associations will increase the number of correspondences because additional match candidates are considered. This extension is also beneficial to handle different concept granularities. For the

Table 2: Match results for the baseline matcher

	# Concepts using direct associations	# Concepts using indirect associations
# Correspondences	711	2,251
# Matched Softunity concepts	132 (28.1%)	160 (34.0%)
$MatchRatio_{SU}$	5.4	14.1
# Matched Amazon concepts	339 (18.3%)	364 (19.6%)
$MatchRatio_{AM}$	2.1	6.2

starting example in Figure 1, indirect associations can help match the *Operating Systems* concepts, although the Amazon concept has no directly associated products.

Table 2 indicates that the baseline matcher finds correspondences only for a minority of the concepts, namely 28% (34%) of the Softunity and 18% (20%) of the Amazon concepts using direct (indirect) associations. The match ratios are rather high; using indirect associations almost triples the match ratios, i.e. the number of matching concepts per matched concept.

Table 3 confirms that *dice similarity* is very restrictive making it difficult to obtain high concept similarities. Hence only few correspondences are achieved for direct associations and only few concepts can be matched (low recall). As shown in Fig. 2, for all similarity thresholds the match coverage is less than 30% compared to the baseline matcher. On the other hand, the quality of the correspondences is quite good. For example, with a 50% similarity threshold we obtain 71 correspondences covering 60 (68) different Softunity (Amazon) concepts leading to a very good match ratio of 1.2 (1.0). The baseline approach, on the other hand, uses the ten-fold number of correspondences for matching about twice the number of Softunity concepts (ratio 5.4) and five times the number of Amazon concepts (ratio 2.1). Indirect associations help to slightly improve the match coverage for dice without impairing the match ratios. In section 4 we analyze how the match coverage can be further extended by considering set correspondences.

The *minimum similarity* metric is less restrictive than dice similarity and determines many more correspondences. Furthermore, many more concepts can be matched (Figure 2) so that match coverage is improved significantly for our test data. Even for a similarity threshold of 1 (100%) a match coverage of up to 80% is achieved. This good coverage is obtained with many fewer correspondences than in the baseline case (ratios of about 2.7 for Softunity and 1.1 for Amazon). Compared to dice similarity the much improved recall is achieved with a similar good precision for Amazon concepts. The higher ratio for Softunity is influenced by the much higher number of Amazon concepts so that more correspondences are needed per Softunity concept to match most instances. In

Table 3: Number of concept correspondences for instance-based matching

		Similarity Threshold						*Baseline*
Association	*Metric*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*	
Direct	Dice	71	40	21	17	13	11	711
	Min	389	308	255	233	213	208	
Indirect	Dice	90	62	34	30	23	12	2.251
	Min	500	425	385	364	346	335	

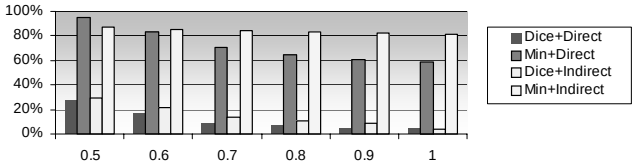


Figure 2: Match coverage (w.r.t. the baseline matcher) for instance-based matching and different similarity thresholds

summary, using the minimum similarity is the best match approach for the considered e-commerce scenario and more appropriate than dice.

3.4 Comparison between metadata- and instance-based matching

To compare the instance-based approaches with metadata-based ontology matching we applied different name matchers on the product catalogs. Several name-based mappings are determined by using the trigram string similarity between the concept names of Amazon and Softunity. The mapping NAME-SU determines for each Softunity (SU) concept the Amazon concept with the most similar name; a correspondence is only assumed if the similarity values exceeds a minimal similarity of 80%. The mapping NAME-AM analogously determines the correspondences for Amazon (AM) concepts. The symmetrical mapping NAME-SUAM only selects correspondences fulfilling a “stable marriage”, i.e., the best matching Amazon concept for a given Softunity concept has the same Softunity concept as the best match, too. Three additional name mappings are determined which concatenates the concept names with the names of all parent concepts (Path matcher). This way names become less ambiguous and reflect the structural position of a concept within the ontology. Due to the high diversity of path names we use the best correspondences for each Softunity (Path-SU) and each Amazon (Path-AM) concept respectively without checking for a minimal similarity value. Similar to the name matcher Path-SUAM only selects correspondences fulfilling a “stable marriage”.

Table 4 summarizes our results. The first observation is that the simple name matchers match relatively few concepts (31% for Softunity; 9% for Amazon) but determine correspondences with a rather high match ratio (4.0 – 4.7). The reason is that many concepts have equal or similar names (e.g., "*miscellaneous*") but are not related to each other. This ambiguity is reduced when using the path name instead of concept name only. The symmetrical path matcher Path-SUAM seems most successful as it achieves a perfect match ratio of 1 for both ontologies. Moreover, Path-SUAM achieves a comparable number of matched concepts than the name matchers but with only a fraction of correspondences.

Table 2: Match results for metadata-based matching approaches

Matcher	# Correspondences	# Matched SU concepts	# Matched AM concepts	Match Ratio SU	Match Ratio AM
Name-SU	696	148 (31.5%)	174 (9.4%)	4.7	4.0
Name-AM	695	147 (31.3%)	174 (9.4%)	4.7	4.0
Name-SUAM	695	147 (31.3%)	174 (9.4%)	4.7	4.0
Path-SU	492	470 (100.0%)	205 (11.0%)	1.0	2.4
Path-AM	1,881	262 (55.7%)	1,856 (100.0%)	7.2	1.0
Path-SUAM	155	155 (33.0%)	153 (8.2%)	1.0	1.0

Comparing the number of matched concepts of the baseline approach (Table 2) with the metadata approaches (Table 4) we see a similar match coverage for Softunity. On the other hand, the metadata-based approaches match only half of the Amazon concepts (with the exception of Path-AM). However, a similar number of matched concepts does not mean that the same concepts are matched by the different approaches. We therefore determine the overlap of the metadata-based and instance-based matching using the baseline scheme as well as the dice and minimal similarity metrics (similarity threshold of 50%). Table 5 shows the number of shared correspondences for the different approaches. For example, the Path-SU matcher determines 492 correspondences whereas the instance based matcher using the dice similarity metric and direct associations determines 71 correspondences. But only 20 correspondences can be found in both match results.

Table 5 reveals a very small correspondence overlap between the metadata-based and instance-based matchers for both direct and indirect associations. The path matchers return a much higher overlap than the name matchers underlining their superiority. The highest relative overlap is achieved for Path-SUAM for which almost 30% of the correspondences are also obtained by the baseline instance matcher. For the instance-based matchers the dice similarity metric obtains the smallest overlap, while the minimum similarity achieves about 80% as many overlapping correspondences as the baseline matcher. Interestingly, for the minimum similarity there is hardly any difference in the overlap between direct and indirect associations although the latter generates significantly more correspondences. The results show that the metadata-based matching approaches miss many concept correspondences with a significant instance overlap. On the other hand, name-based matching identifies many correspondences without instance overlap. Note that these correspondences are not necessarily wrong but can be useful to

Table 3: Overlap of metadata and instance-based ontology matching approaches

		Baseline		Dice		Min	
		Direct	Indirect	Direct	Indirect	Direct	Indirect
		711	2,251	71	90	389	500
Name-SU	696	13	15	5	7	10	13
Name-AM	695	13	15	5	7	10	13
Name-SUAM	695	13	15	5	7	10	13
Path- SU	492	54	62	20	23	45	44
Path-AM	1,881	109	132	24	34	92	92
Path-SUAM	155	41	47	14	17	35	34

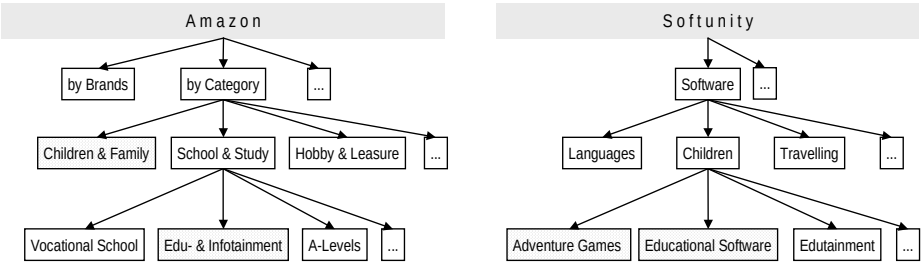


Figure 3: Portions of two application-specific ontologies with related concepts

find related products even in the absence of matching instances, e.g. when stores have similar but different products (e.g. equivalent products from a different manufacturer). Altogether the experiment clearly shows the need for both approaches, instance- and metadata-based matching.

4 Set Correspondences

The correspondences considered so far related single concepts. Set correspondences relate sets of concepts between two ontologies. We motivate the use of set correspondences, explain their calculation and evaluate them for our test data. Throughout this section we focus on the restrictive dice similarity and direct associations which were shown to determine high quality correspondences but need recall improvements to match more concepts.

4.1 Motivating example

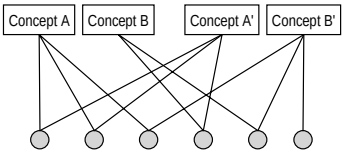
Figure 3 illustrates that set correspondences may express semantic relationships better than single correspondences. For example, we assume that none of the two highlighted Softunity concepts (*Adventure Games*, *Educational Software*) corresponds to only one of the highlighted Amazon concepts (*Children & Family*, *Edu- & Infotainment*). Hence to accurately describe such a N:M relationship between concepts we should be able to use one correspondence between concept sets rather than only correspondences between single concepts.

We therefore generalize the dice similarity for set correspondences. Given two concept sets C_1 and C_2 as subsets of all concepts C_{O1} and C_{O2} of two ontologies we define

$$Sim_{DICE}(C_1, C_2) = \frac{2 \cdot |I_{C_1} \cap I_{C_2}|}{|I_{C_1}| + |I_{C_2}|} \in [0...1], \forall C_1 \subseteq C_{O1}, C_2 \subseteq C_{O2}$$

Analogously, I_{C1} and I_{C2} are the union sets of associated instances to concept sets C_1 and C_2, respectively, whereas $I_{C1} \cap I_{C2}$ denotes the matching instances to both concept sets.

Figure 4 illustrates the use of the generalized dice similarity metric for a more abstract example with two matching concept pairs {A, B} and {A', B'}. The circles denote instances that are associated with concepts. For example, the left-most instance (circle) is assumed to be associated to both concepts A and A'. The computation of the instance-based dice similarity for single correspondences leads to the result given in the table



	A’	B’
A	2*2/(3+3) = 0.67	2*1/(3+3) = 0.33
B	2*1/(2+3) = 0.4	2*1/(2+3) = 0.4

Figure 4: Example for computation of the generalized dice similarity

(assuming cardinalities 3, 2, 3, and 3 for concepts A, B, A’ and B’, respectively). On the other hand, the generalized dice similarity for the set correspondence {A, B}-{A’, B’} is 2*5/(5+6) ≈ 0.9 and therefore higher than for all considered single correspondences. The example demonstrates that set correspondences may have much higher similarity values (instance overlaps) than single concept correspondences and are therefore useful for representing relationships between concepts.

4.2 Determining Set Correspondences

Set correspondences are established during an iterative process based on the single correspondences that are a special case of set correspondences. Concepts are successively added to the sets on both sides of the correspondence. It is important to note that the extension of a concept set by one concept must improve the correspondence similarity to avoid trivial set correspondences. Therefore no concepts are added that do not strengthen the correspondence. Hence, we require that for all concept sets A and B it holds:

$$A' \subseteq A \ \wedge \ B' \subseteq B \ \wedge \ (A' \neq A \ \vee \ B' \neq B) \ \rightarrow \ \text{Similarity (A-B)} \ > \ \text{Similarity (A'-B')}$$

4.3 Experimental evaluation

In the following experiment we start from the single correspondences using direct associations and the dice similarity metric. We generate concept sets step-by-step up to a maximum of three concepts per set and count the number of resulting correspondences with at least 50% similarity. Table 6 shows the number of correspondences w.r.t. the size of the concept sets, e.g., we count 30 correspondences between sets of two Softunity concepts and one Amazon concept.

The comparison of Softunity and Amazon shows a different development for the number of correspondences when extending the concept sets. The number of new correspondences increases when considering more Amazon concepts but decreases for Softunity. One reason is that Amazon has many more concepts so that the associated products of one Softunity concept are distributed over multiple Amazon concepts.

The example of Section 4.1 illustrates that set correspondences may involve concepts

		Number of Amazon concepts		
		1	2	3
Number of Softunity concepts	1	71	169	642
	2	30	164	996
	3	16	133	862

Table 4: Number of correspondences

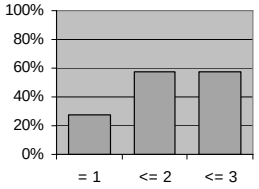
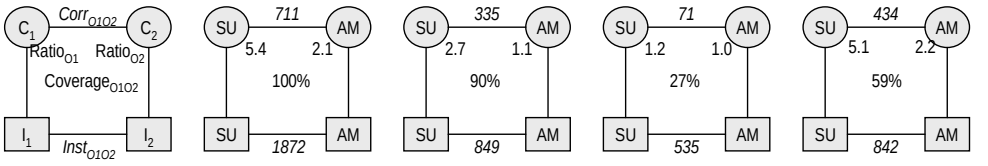


Figure 5: Match coverage



Legend a) Baseline b) Min (100%) c) Dice (50%) d) Dice-Set2 (50%)

Figure 6: Comparison of different matching strategies

that are not present in single correspondences. For example, all single correspondences for concept B are below the 50% threshold, but nevertheless B occurs in the set correspondence {A,B}-{A',B'}. We therefore analyze the number of concepts that participate in set correspondences. Figure 5 shows the match coverage w.r.t. to the maximum number of concepts in the set correspondences. For example, there is almost a 60% coverage for correspondences between sets of one or two Softunity and Amazon concepts. We observe that extending 1:1 matches to sets of two concepts significantly improve the match coverage. Considering larger sets of three concepts, however, only leads to many more correspondences without covering significantly more concepts. Hence match precision is reduced so that – in the considered scenario – set correspondences should be confined to sets of two concepts per ontology.

5 Conclusions & Future Work

We showed that instance matching can effectively be used for matching hierarchical ontologies such as product catalogs. Instance-based matching considers the extensional overlap of concepts and is thus able to find concept correspondences even in the presence of high degrees of semantic heterogeneity, e.g. different concept names or incomparable categorizations. Our experimental evaluation demonstrated the value of the instance-based approach over metadata-based matching which missed many correspondences between concepts sharing the same instances.

To summarize our results for instance-based matching Figure 6 illustrates the number of found concept correspondences (e.g., 335 for the minimum similarity approach using a 100% threshold) as well as the number of used instance correspondences (849) for different match strategies. Fig. 6 also presents the match coverage (90%) as well as the match ratios for Softunity and Amazon concepts (2.7 and 1.1). The comparison indicates the high usefulness of the *minimum similarity* and the recall improvements using *concept sets*. We showed that the minimum similarity nearly achieves the same coverage (Fig. 6b) like the baseline approach (Fig. 6a). Moreover, this coverage is achieved by less than 50% concept correspondences (335 of 711) resulting in much improved match ratios. Comparing Figure 6c and 6d illustrates that set correspondences are able to match significantly more concepts by using a higher number of instance correspondences (535 vs. 842). This underlines our assumption that 1:1 correspondences are often not sufficient for matching ontologies. However, the number of concept correspondences increases as well resulting in rather poor match ratios, e.g. compared to the minimum approach (Fig. 6b). This suggests that many set correspondences do not actually improve match coverage because they only combine already matched concepts.

In future work we will therefore further investigate set correspondences to eliminate such useless set correspondences and improve precision. Furthermore, we plan to apply instance-based matching in different domains, such as life sciences. We also want to further analyze possible combinations of instance- and metadata-based ontology matching.

References

[ADMR05] D. Aumueller, H. Do, S. Massmann, E. Rahm: Schema and ontology matching with COMA++. Proc. of the Intl. Conference on Management of Data (SIGMOD), 2005.

[AGY05] A. Avesansi, F. Giunchiglia, M. Y. Yatskevich: A large taxonomy mapping evaluation. Proc. of the 4th Intl. Semantic Web Conference (ISWC), 2005.

[AS01] R. Agrawal, R. Srikant: On integrating catalogs. Proc of the 10th Intl. World Wide Web Conference (WWW), 2001.

[BAB05] O. Bodenreicher, M. Aubery, A. Burgun: Non-lexical approaches to identifying associative relations in the Gene Ontology. Proc. of the 10th Pacific Symposium on Biocomputing (PSB), 2005.

[BN05] A. Bilke, F. Naumann: Schema matching using duplicates. Proc. of the 21st Intl. Conference on Data Engineering (ICDE), 2005.

[CCL03] C. Chua, R. Chiang, E.-P. Lim: Instance-based attribute identification in database integration. The VLDB Journal, 12(3):228-243, 2003.

[CGL01] D. Calvanese, G. De Giacomo, M. Lenzerini: Ontology of integration and integration of ontologies. Proc. of the Intl. Description Logics Workshop, 2001.

[DH05] A. Doan, A. Halevy: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine 26(1): 83-94, 2005

[DMD+03] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, A. Halevy: Learning to match ontologies on the semantic web. The VLDB Journal 12(4): 303-319, 2003.

[ELT+03] J. Euzenat, D. Loup, M. Touzani, P. Valtchev: Ontology Alignment with OLA. Proc. of the 3rd Intl. Workshop on Evaluation of Ontology-based Tools, 2004.

[ES04] M. Ehrig, S. Staab: QOM – Quick ontology mapping. Proc. of the 3rd Intl. Semantic Web Conference (ISWC), 2004.

[HYN+04] T. Hoshiai, Y. Yamane, D. Nakamura, H. Tsuda: A semantic category matching approach to ontology alignment. Proc of the 3rd Intl. Workshop on Evaluation of Ontology-based Tools, 2004.

[ITH03] R. Ichise, H. Takeda, S. Honiden: Integrating multiple internet directories by instance-based learning. Proc of the 18th Intl. Joint Conference on Artificial Intelligence (IJCAI), 2003.

[KS03] Y. Kalfoglou, M. Schorlemmer: Ontology mapping: The state of the art. The Knowledge Engineering Review Journal, 18(1): 1-31, 2003.

[MB04] P. Mork, P. Bernstein: Adapting a generic match algorithm to align ontologies of human anatomy. Proc. of the 20th Intl. Conference on Data Engineering (ICDE), 2004.

[MS02] A. Maedche, S. Staab: Measuring similarity between ontologies. Proc. of the 13th Conf. on Knowledge Engineering and Management, 2002.

[MTM06] S. Myhre, H. Tveit, T. Mollestad, A. Laegreid: Additional Gene Ontology structure for improved biological reasoning. Bioinformatics 22(16): 2020-2027, 2006.

[NM03] N. Noy, M. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. Intl. Journal of Human-Computer Studies, 59(6):983-1024, 2003.

[PE95] M. Perkowitz, O. Etzioni: Category translation: Learning to understand information on the internet. Proc. of the 14th Intl. Joint Conference on Artificial Intelligence (IJCAI), 1995.

[Rijs79] C. J. van Rijsbergen. *Information retrieval*. Butterworths, London, 2nd edition, 1979.

[RB01] E. Rahm, P. Bernstein: A survey of approaches to automatic schema matching. The VLDB Journal 10(4): 334-350 , 2001.

[SE05] P. Shvaiko, J. Euzenat: A survey of schema-based matching approaches. Journal on Data Semantics, LNCS 3730 (JoDS IV): 928-943, 2005.

[WVV+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner: Ontology-based integration of information – A survey of existing approaches. Proc. of the IJCAI Workshop on Ontologies and Information Sharing, 2001.