Semantic Enrichment of Ontology Mappings: A Linguistic-based Approach

Patrick Arnold, Erhard Rahm

University of Leipzig, Germany

Abstract. There are numerous approaches to match or align ontologies resulting into mappings specifying semantically corresponding ontology concepts. Most approaches focus on finding equality correspondences between concepts, although many concepts may not have a strict equality match in other ontologies. We present a new approach to determine more expressive ontology mappings supporting different kinds of correspondences such as equality, is-a and part-of relationships between ontologies. In contrast to previous approaches, we follow a so-called enrichment strategy that semantically refines the mappings determined with a state-of-the art match tool. The enrichment strategy employs several linguistic approaches to identify the additional kinds of correspondences. An initial evaluation shows promising results and confirms the viability of the proposed enrichment strategy.

1 Introduction

There are numerous approaches and tools for ontology matching or alignment, i.e., the automatic or semi-automatic identification of semantically corresponding or matching concepts in related ontologies [14], [2]. These approaches typically utilize different techniques exploiting the linguistic and structural similarity of concepts and their neighborhood or the similarity of concept instances. All determined correspondences between two ontologies build a so-called ontology mapping. Ontology mappings are useful for many tasks, e.g., to merge related ontologies or to support ontology evolution.

A restriction of most previous match approaches is that they focus on finding truly matching pairs of concepts so that each correspondence expresses an equality relationship between two concepts. This is a significant limitation, since a more expressive mapping should also include further kinds of correspondences, such as is-a or part-of relationships between concepts. Such more expressive or semantic mappings are generally beneficial and have been shown to substantially improve ontology merging [11]. The existing approaches have even problems with finding truly equivalent concepts, since similarity-based match approaches are inherently approximative, e.g., if one assumes a match and the concept names have a string similarity above some threshold. Hence, the correspondences often express only some "relatedness" between concepts that can reflect equality or some weaker (e.g. is-a) relationship.

For illustration, we have shown in Figure 1 (left) the result for matching two simple ontologies with the state-of-the-art match tool COMA 3.0 (the successor of COMA++) [1], [9]. Each line represents a correspondence between two concepts. The example shows that not all such correspondences represent equality relationships, e.g., *Action_Games - Games*.

We present a new approach to determine more expressive ontology mappings supporting different kinds of correspondences such as equality, is-a and part-of relationships between ontologies. There are already a few previous approaches to identify such mappings (see Section 2), but they are still far from perfection. They have in common that they try to directly identify the different kinds of relationships, typically with the help of dictionaries such as WordNet. By contrast, we propose a so-called enrichment strategy implementing a two-step approach leveraging the capabilities of state-of-the art match tools. In a first step we apply a common match tool to determine an initial ontology mapping with approximate equality correspondences. We then apply different linguistic approaches (including the use of dictionaries) to determine for each correspondence its most likely kind of relationship. In Figure 1 (right) we illustrate how the enrichment approach can improve the mapping by identifying several is-a and inverse is-a relationships. The two-step approach has the advantage that it can work in combination with different match tools for step 1, and that it has to process relatively compact mappings instead of evaluating a large search space as for 1-step semantic match approaches. As we will see in the evaluation, we can still achieve a high match effectiveness.

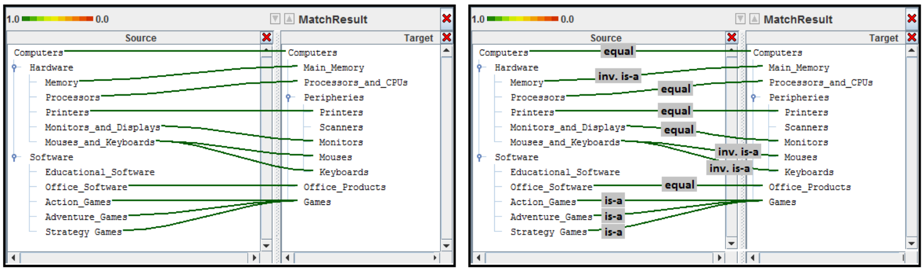

Fig. 1: Input (left) and output (right) of the Enrichment Engine

Our contributions are as follows:

- We propose a new two-step, enrichment approach to semantic ontology matching that refines the correspondences of an equality-based ontology mapping by different kinds of relationships between concepts (Section 3).
- We propose the combined use of four linguistic-based approaches for determining the relationship type of correspondences, including the use of background knowledge such as dictionaries (Section 4).
- We evaluate the new approach for different real-life test cases and demonstrate its high effectiveness (Section 5).

2 Related Work

Only a few tools and studies already try to determine different kinds of correspondences or relationships for ontology matching. S-Match [4][5] is one of the first such tools for "semantic ontology matching". They distinguish between equivalence, subset (is-a), overlap and mismatch correspondences and try to provide a relationship for any pair of concepts of two ontologies by utilizing standard match techniques and background knowledge from WordNet. Unfortunately, the result mappings tend to become very voluminous with many correspondences per concept while users are normally interested only in the most relevant ones. We tried to apply S-Match to our evaluation scenarios and report on the results in Section 5.

Taxomap [7] is an alignment tool developed for the geographic domain. It regards the correspondence types equivalence, less/more-general (is-a / inverse is-a) and is-close ("related"). It uses linguistic techniques and background sources such as WordNet. The linguistic strategies seem rather simple; if a term appears as a part in another term, a more-general relation is assumed which is not always the case. For example, in Figure 1 the mentioned rule holds for the correspondence between *Games* and *Action_Games* but not between $Monitors$ and $Monitors\_and\_Displays$. In [12], the authors evaluated Taxomap for a mapping scenario with 162 correspondences and achieved only a low recall of 23 % and a good precision of 89 %.

Several further studies deal with the identification of semantic correspondence types without providing a complete tool or framework. An approach utilizing current search engines is introduced in [6]. For two concepts A, B they generate different search queries like "A, such as B" or "A, which is a B" and submit them to a search engine (e.g., Google). They then analyze the snippets of the search engine results, if any, to verify or reject the tested relationship.

The approach in [13] uses the Swoogle search engine to detect correspondences and relationship types between concepts of many crawled ontologies. The approach supports equal, subset or mismatch relationships. [15] exploits reasoning and machine learning to determine the relation type of a correspondence, where several structural patterns between ontologies are used as training data.

3 Overview and Workflow

An ontology O consists of a set of concepts C and relationships R, where each $r \in R$ links two concepts $c_1, c_2 \in C$. In this paper, we assume that each relation in O is either of type "is-a" or "part-of". We call a concept *root* if there is no other concept linking to it. A path from a root to a concept is called a *concept path*. We denote concept paths as follows: $root.concept_1.concept_2.(...).concept_n$. Each concept is referenced by its *label*.

A correspondence C between two ontologies O_1 and O_2 consists of a source concept $C_S \in O_1$, a target concept $C_T \in O_2$, a relationship or correspondence type, and an optional confidence value between 0 and 1 expressing the computed

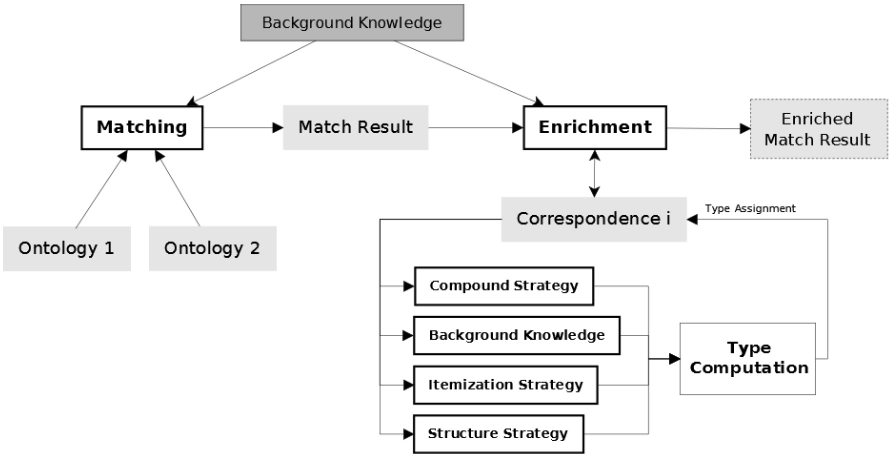

Fig. 2: Basic Workflow for Mapping Enrichment

likelihood of the correspondence. In this study, we consider six correspondence types: equal, is-a (or subset), inverse is-a, part-of (or composition), has-a (inverse of part-of) and related.

The basic workflow of our enrichment approach is shown in Figure 2. It consists of two steps: (initial) matching and enrichment. The matching step is performed using a common tool for ontology matching. It takes two ontologies and possibly additional background knowledge sources as input and computes an initial match result (a set of correspondences). This match result together with background knowledge sources is the input for the enrichment step. In the enrichment step, we currently apply four strategies (Compound, Background Knowledge, Itemization, Structure) for each input correspondence to determine the correspondence type. The four strategies will be described in the next section. Each strategy returns one type per correspondence or "undecided" if no type can be confirmed. From the individual results gained by the four strategies, we determine the final type and assign it to the correspondence (we apply the type which was most frequently returned by the strategies). Our final match result consists of the semantically enriched correspondences with an assigned relationship type.

The four strategies may return different correspondence types or only "undecided". In case that all strategies return "undecided", we will apply the equal type, because it is the default correspondence type from the initial ontology matching step. If there are different outcomes from the strategies (e.g., one strategy decided on equal, one on is-a and the other two returned undecided), we have different possibilities to decide on one type. We could either prioritize the relationship types or the strategies. For the latter we can use the experienced degree of effectiveness of the different strategies based on evaluation results. We could also interact with the user to request a manual decision about the correspondence type. Currently, we use the latter option, although contradicting decisions occurred extremely rarely in our tests.

Our two-step approach for semantic ontology matching offers different advantages. First of all, we reduce complexity compared to 1-step approaches that try to directly determine the correspondence type when comparing concepts in O_1 with concepts in O_2. For large ontologies, such a direct matching is already time-consuming and error-prone for standard matching. The proposed approaches for semantic matching are even more complex and could not yet demonstrate their general effectiveness. Secondly, our approach is generic as it can be used for different domains and in combination with different matching tools for the first step. On the other hand, this can also be a disadvantage since the enrichment step depends on the completeness and quality of the initially determined match result. Therefore, it is important to use powerful tools for the initial matching and possibly to fine-tune their configuration. In our evaluation, we will use the COMA match tool that has already shown its effectiveness in many domains [9].

4 Implemented Strategies

In the following we will introduce the four implemented strategies to determine the correspondence type. Table 1 gives an overview of the strategies and the relationship types they are able to detect. It can be seen that the Background Knowledge approach is especially valuable as it can help to detect all relationship types. All strategies are able to identify is-a correspondences.

Strategy	equal	is-a	part-of	related
Compounding		X		
Background K.	X	X	X	X
Itemization	X	X		
Structure		X	X	

Table 1: Supported correspondence types per strategy

4.1 Compound Strategy

In linguistics, a compound is a special word W that consists of a head W_H carrying the basic meaning of W, and a modifier W_M that specifies W_H [3]. In many cases, a compound thus expresses something more specific than its head, and is therefore a perfect candidate to discover an is-a relationship. For instance, a blackboard is a board or an apple tree is a tree. Such compounds are called *endocentric compounds*. There are also *exocentric compounds* that are not related with their head, such as buttercup, which is not a cup, or saw tooth, which is not a tooth. These compounds are of literal meaning (metaphors) or changed their spelling as the language evolved, and thus do not hold the is-a relation, or only to a very limited extent (e.g., airport, which is a port only in a broad sense). There is a third form of compounds, called *appositional* or *copulative* compounds, where the two words are at the same level, and the relation is rather more-general (inverse is-a) than more-specific, as in Bosnia-Herzegowina, which means both Bosnia and Herzegowina, or bitter-sweet, which means both bitter and sweet (not necessarily a "specific bitter" or a "specific sweet"). However, this type is quite rare.

In the following, let A, B be the literals of two concepts of a correspondence. The Compound Strategy analyzes whether B ends with A. If so, it seems likely that B is a compound with head A, so that the relationship B is-a A (or A inv. is-a B) is likely to hold. The Compound approach allows us to identify the three is-a correspondences shown in Figure 1 (right).

We added an additional rule to this simple approach: B is only considered a compound to A if $length(B) - length(A) \geq 3$, where $length(X)$ is the length of a string X. Thus, we expect the supposed compound to be at least 3 characters longer than the head it matches. This way, we are able to eliminate obviously wrong compound conclusions, like *stable* is a *table*, which we call *pseudo compounds*. The value of 3 is motivated by the observation that typical nouns or adjectives consist of at least 3 letters.

We also tested a variation of the approach where we extracted the modifier of a supposed compound and checked whether it appears in a word list or dictionary. This is expected to prevent pseudo compounds like "*nausea* is a *sea*". We found that this approach does not improve our results, so we do not consider it further.

4.2 Background Knowledge Strategy

The use of background knowledge such as thesauri is a powerful approach since it can provide many linguistic relationships between words that are helpful to determine different relationships between concepts. Table 2 summarizes different linguistic relationships with typical examples as well as their associated kind of correspondence.

For preciseness, we briefly characterize the different linguistic relationships between words [10]. Two words $X \neq Y$ of a language are called *synonyms* if they refer to the same semantic concept, that is, if they are similar or equivalent in meaning. They are called *antonyms* if they are different in meaning (in the broad sense) or describe opposite or complementary things (in the narrow sense). X is a *hypernym* of Y if it describes something more general than Y. Y is then called the *hyponym* of X. X is a direct hypernym of Y if there is no word Z so that Z is a hypernym of Y and X is a hypernym of Z.

Linguistic relationship	**Example**	**Corresp. type**
Synonyms	river, stream	equal
Antonyms	valley, mountain	mismatch
Hypernyms	vehicle, car	inv. is-a
Hyponyms	apple, fruit	is-a
Holonyms	body, leg	has-a
Meronyms	roof, building	part-of
Cohyponyms	oak, maple, birch	related

Table 2: Typical linguistic and semantic relationships

X and Y are cohyponyms if there is a concept Z which is the direct hypernym of X and Y. X is a *holonym* of Y if a "typical" Y is usually part of a "typical"

X. The expression "typical" is necessary to circumvent special cases, like cellar is part of house (there are houses without a cellar, and there are cellars without a house). X is then called the *meronym* of Y.

In our current implementation of the Background Knowledge Strategy we use WordNet 3.0 [18] to determine the semantic relationship between two concepts of an input correspondence. We used the Java API for WordNet Searching (JAWS) [17] to retrieve information from WordNet, and implemented an interface to directly answer queries like "Is X a (direct) hypernym of Y?". We observed that WordNet is a very reliable source, which is able to detect many non-trivial relationships that cannot be detected by other strategies.

In case that an open compound C matches a single word W, where W is found in WordNet, yet C is not, we gradually remove the modifiers of C in order to detect the relationship. After each reduction, we check whether this form is in WordNet, and if not, proceed till we reach the head of C. For instance, we encountered correspondences such as ("US Vice President", "Person"), where "US Vice President" was not in the dictionary. However, "Vice President" is in the dictionary, so after the first modifier removal, WordNet could return the correct type (is-a).

4.3 Itemization Strategy

The itemization strategy is used if at least one of the two concepts in a correspondence is an itemization. We define an itemization as a list of items, where an item is a word or phrase that does not contain commas, slashes or the words "and" and "or". We call concepts containing only one item *simple concepts*, like "Red Wine", and concepts containing more than one item *complex concepts*, like "Champagne and Wine".

Itemizations need a different treatment than simple concepts, because they contain more information than a simple concept. Regarding itemizations also prevents us from detecting pseudo compounds, like "bikes and cars", which is not a specific form of cars, but something more general. Hence, there is in general an inverse is-a relationship between itemizations and the items they contain, e.g., between "cars and bikes" and cars resp. bikes. Two inv. is-a correspondences shown in Figure 1(right) are based on such itemizations (e.g., mouses and keyboards). Our itemization strategy is not restricted to such simple cases, but also checks whether there are is-a relationships between the items of an itemization. This is necessary to find out, for example, that "computers and laptops" is equivalent to a concept "computer", since laptop is just a subset of computer.

We now show how our approach determines the correspondence types between two concepts C_1, C_2 where at least one of the two concepts is an itemization with more than one item. Let I_1 be the item set of C_1 and I_2 the item set of C_2. Let w_1, w_2 be two words, with $w_1 \neq w_2$. Our approach works as follows:

1. In each set I remove each $w_1 \in I$ which is a hyponym of $w_2 \in I$.
2. In each set I, replace a synonym pair $(w_1 \in I, w_2 \in I)$ by w_1.
3. Remove each $w_1 \in I_1$, $w_2 \in I_2$ if there is a synonym pair (w_1, w_2).

4. Remove each $w_2 \in I_2$ which is a hyponym of $w_1 \in I_1$.
5. Determine the relation type:
   (a) If $I_1 = \emptyset, I_2 = \emptyset$: equal
   (b) If $I_1 = \emptyset, |I_2| \geq 1$: is-a
   (c) If $|I_1| \geq 1, I_2 = \emptyset,$: inverse is-a
   (d) If $|I_1| \geq 1, I_2 \geq 1$: undecided

The rationale behind this algorithm is that we remove items from the item sets as long as no information gets lost. Then we compare what is left in the two sets and come to the conclusions presented in step 5.

Let us consider the concept pair $C_1 =$ "books, ebooks, movies, films, cds" and $C_2 =$"novels, cds". Our item sets are $I_1 = \{books, ebooks, movies, films, cds\}$, $I_2 = \{novels, cds\}$. First, we remove synonyms and hyponyms within each set, because this would cause no loss of information (steps 1+2). We remove $films$ in I_1 (because of the synonym *movies*) and *ebooks* in I_1, because it is a hyponym of *books*. We have $I_1 = \{books, movies, cds\}, I_2 = \{novels, cds\}$. Now we remove synonym pairs between the two item sets, so we remove *cds* in either set (step 3). Lastly, we remove a hyponym in I_1 if there is a hypernym in I_2 (step 4). We remove *novel* in I_2, because it is a *book*. We have $I_1 = \{books, movies\}, I_2 = \emptyset$. Since I_1 still contains items, while I_2 is empty, we conclude that I_1 specifies something more general, i.e., it holds C_1 inverse is-a C_2.

If neither item set is empty, we return "undecided" because we cannot derive an equal or is-a relationship in this case.

4.4 Structure Strategy

The structure strategy takes the explicit structure of the ontologies into account. For a correspondence between concepts Y and Z we check whether we can derive a semantic relationship between a father concept X of Y and Z (or vice versa). For an is-a relationship between Y and X we draw the following conclusions:

– X equiv $Z \rightarrow Y$ is-a Z
– X is-a $Z \rightarrow Y$ is-a Z

For a part-of relationship between Y and X we can analogously derive:

– X equiv $Z \rightarrow Y$ part-of Z
– X part-of $Z \rightarrow Y$ part-of Z

The approach obviously utilizes the semantics of the intra-ontology relationships to determine the correspondence types for pairs of concepts for which the semantic relationship cannot directly be determined.

For example, consider the correspondence (vehicles.cars.convertibles, vehicles. cars). Let us assume that "convertibles" is not in the dictionary. No other strategy would trigger here. However, it can be seen that the leaf node "cars" of the second concept matches the father of the leaf node in the first concept.

Since "convertibles" is a sub-concept of its father concept "cars", we can derive the is-a relationship for the correspondence.

To decide whether X and Z are equivalent or in an is-a or part-of relationship we exploit three methods: name equivalence (as in the example, cars = cars), WordNet and Compounding, thus exploiting the already implemented strategies.

4.5 Verification Step

We observed that the identification of is-a (subset) correspondences can fail when the concepts are differently organized within hierarchies in the input ontologies. Consider the correspondence ("apparel.children_shoes", "clothing.children. shoes"). Based on the leaf concepts "children_shoes" and "shoes" both the Compound and Background strategies would suggest an "is-a" correspondence, because children shoes are obviously shoes. However, a closer look on the two paths reveals that both concepts are in fact equal.

To deal with such cases we implemented a verification step to post-process presumed is-a correspondences. For this purpose, we combine the leaf concept with the father concept and check whether the combination matches the opposite, unchanged leaf concept of a correspondence. For the above example, the combination of "children" and "shoes" on the target side leads to an equivalence match decision so that the is-a relationship is revoked.

This simple approach already leads to a significant improvement, but still needs extensions to deal with more complex situations such as:

1. The actual meaning is spread across multiple levels, like ("children.footware. shoes", "children shoes").
2. The father node of a concept A may not match the modifier of a corresponding concept B, like ("kids.shoes", "children shoes"). Here, we would have to check whether the father node of A ("kids") is a synonym to the modifier in B ("children").

We plan to deal with such extensions in future work.

5 Evaluation

Evaluating our approach is more difficult than classic ontology matching techniques, because in many cases the true relationship cannot be unequivocally determined. For example, the correspondence type for (street, road) could be considered is-a (as suggested by WordNet) or equal. There might even be different relationships depending on the chosen domain or purpose of the ontologies. Consider the word *strawberry*, which biologically is not a berry but a nut. Thus, in a biological ontology, claiming *strawberry* is a *berry* would be wrong, whereas in a food ontology (of a supermarket etc.) it might be correct, since a customer would expect *strawberries* to be listed under the concept *berries*.

Another difficulty of evaluating the enrichment approach is its dependency on the match result of step 1 that might be incomplete and partially wrong.

No	Domain	Lang.	#Corr.	equal	is-a	has-a	related
G_1	Web Directories	DE	340	278	52	5	5
G_2	Diseases	EN	395	354	40	1	0
G_3	TM Taxonomies	EN	762	70	692	0	0

Table 3: Overview of the Gold Standards

Furthermore, the evaluation may consider all relationship types or only the ones different from equality.

For our evaluation, we use three test cases for which we manually determined the presumed perfect match result (Gold Standard) with semantic correspondence types. After the representation of these cases, we will first evaluate our approach under best-case conditions by providing it with all correspondences (without relationship types) of the Gold Standard. We then use the state-of-the art match tool (COMA 3.0) to determine an approximative initial match result for enrichment. We also report on how the 1-step semantic approach of S-match performed for our test cases. Finally, we summarize observations on the four individual strategies applied during enrichment.

5.1 Evaluation Scenarios

For our evaluation, we used three ontology matching scenarios of different domains and complexity with manually defined Gold Standards G_1, G_2, G_3. Table 3 provides key information about these standards, such as the domain, language (German, English) as well as the total number of correspondences and their distribution among the different semantic types (equal/is-a/has-a/related).

G_1 is a mapping between the Yahoo and Google Web taxonomies (product catalogs of shopping platforms), consisting of 340 correspondences. The ontologies are in German language, so WordNet has no impact on the result. This scenario contains many itemizations, which the other scenarios lack. G_2 is an extract of 395 correspondences between the diseases catalogs of Yahoo and dmoz. The ontologies are quite domain-specific (medical domain). G_3 is the largest mapping and based on the text mining (TM) taxonomies OpenCalais and AlchemyAPI. It was created and provided by SAP Research and consists of about 1,600 correspondences. In this scenario, half of the correspondences were of the type "related". We noticed significant problems with these correspondences, since many of them were actually of type is-a or has-a or even mismatches. We thus decided to ignore all "related" correspondences in G_3 leaving us with 762 correspondences of type equal, is-a and inverse is-a.

5.2 Tests against Gold Standards

We first evaluate our approach against the manually defined Gold Standards. The input was the Gold Standard containing only the correspondences, the output

	r	p	f
G_1	.467	.690	.578
G_2	.585	.800	.692
G_3	.654	.977	.811

(a) Non-equal types

	r	p	f
G_1	.953	.889	.921
G_2	.982	.947	.964
G_3	.942	.213	.577

(b) Equal-types

	r	p	f
G_1	.899	.864	.881
G_2	.951	.941	.946
G_3	.684	.675	.679

(c) Overall result

Table 4: Evaluation against benchmark

was the Gold Standard with a relation type annotation on each correspondence.

Table 4 shows the recall / precision and f-measure results for the three scenarios. Table **a)** only evaluates the non-equal types, which is of particular interest as such correspondences cannot be identified by standard matching approaches. It shows that the enrichment approach achieves a high f-measure of 58 to 81%, indicating a very good effectiveness. Precision was especially good (69 to 98%) while recall was somewhat limited.

Table **b)** only considers the equal-type. In the first and second scenario, the equal relationship dominates (about 90 % of all correspondences) and in these cases both recall and precision are very high. By contrast, in the third scenario we achieve only a poor precision and medium recall and f-measure. This is influenced by our policy that we denote the equal-type if no other type can be verified which is relatively often the case for the third scenario. We observe that the results for non-equal correspondences in **a)** and for equal correspondences in **b)** are inversely interrelated. For G_3, we achieved the best effectiveness for non-equal correspondences but the lowest for equal correspondences.

Finally, Table **c)** summarizes the overall results considering all correspondence types. F-measure values range from 68 to 95 %, indicating a high effectiveness of the proposed enrichment approach thus demonstrating its viability.

Running these tests took 2.18 s for G_1, 3.41 s for G_2 and 7.47 s for G_3. We thus observed an execution time of 5.5 ms per correspondences in the first scenario and approx. 10.0 ms per correspondence in the second and third scenario where WordNet was effectively used.

5.3 Tests with COMA 3.0

In the second set of experiments we apply the ontology matching tool COMA 3.0 [9] for the initial matching to determine a real, imperfect input mapping for the enrichment step. Since we had no ontologies for G_3, we could not generate a mapping for this scenario and were compelled to restrict the verification to the first two scenarios.

Table 5 shows the results for the COMA-based experiments. Table **a)** shows the quality results for the initial match result where we only checked the recall (completeness) and precision (correctness) of the correspondences generated by COMA (ignoring the correspondence type). The results are relatively low (f-measure between 61 and 72 %) underlining the hardness of the match scenarios.

	r	p	f
G_1	.702	.735	.718
G_2	.673	.543	.608

(a) Quality of initial match result

	r	p	f
G_1	.145	.204	.174
G_2	.365	.441	.403

(b) Results for non-equal types

	r	p	f
G_1	.762	.754	.758
G_2	.703	.547	.625

(c) Results for equal type

	r	p	f
G_1	.669	.680	.674
G_2	.670	.539	.604

(d) Overall results for enrichment

Table 5: Evaluation with COMA match results

Table 5**b)** shows the recall and precision for the detected non-equal types. We knew that the recall of **b)** must be lower than in table 4 **a)**, because in the initial match result some typed correspondences were missing. Still, the recall for G_1 was surprisingly low. By analyzing the result we noticed that COMA aims at a high precision for equal results so that most non-equal results are not retained in its match results. In future work we plan to adjust the COMA settings to reduce this problem.

Table 5**c)** shows the results for the equal correspondences and eventually Table **d)** shows the overall results for all kinds of correspondences. Since most correspondences are of the equal type in G_1 and G_2, most correspondences were correctly typed, and therefore the result in **d)** is only slightly below the result in **a)**.

5.4 Tests with S-Match

For this experiment we tried to apply the latest version of S-Match (s-match-20110422 from 2011) to our evaluation scenarios. It turned out to be very difficult comparing our approach with S-Match, because S-Match practically draws correspondences between each node pair since it also aims at determining mismatches.

For G_1, S-Match returned only 4 match correspondences, which have been all incorrect. This was very surprising despite the fact that the ontologies use German language. This is because there are several trivial correspondences with equal names which any matching tool should be able to detect.

By stark contrast, S-Match returned about 19,600 correspondences for scenario G_2. The root concept "Health" in the first ontology practically corresponded to any concept in the second ontology. This made it difficult to judge the result, because with 395 correspondences in the perfect result, the precision could be at most 2 %. There were 19,563 subset relations and 42 equivalence relations. According to our tests, 12 of the 42 equivalence relations were correct, but none of the subset relations. This would lead to a recall of 3.0 % and a precision being 0.06 %.

We saw that for the problem we addressed, to match taxonomies, S-Match is not a convenient tool. It may be helpful in smaller scenarios or apart from taxonomies, but in our tests it was of no use. While it only returned 4 correspondences in G_1, it returned 19,600 correspondences in G_2, which is hardly possible for any user to verify manually. Apparently S-Match does not rank correspondences to filter out only the most relevant ones.

5.5 Evaluating the Strategies

We also ran our test cases with each single strategy to reveal the strength and weaknesses of our strategies.

Compounding offers a good precision and practically works in all domains, even in different languages (Germanic Languages). Its recall is mostly limited because of the different possibilities how an is-a relation can be expressed. Background Knowledge (WordNet) turned out to be a very precise approach, allowing a precision close to 100 %. However, WordNet only works for the English language and has a limited recall, because of the limited vocabulary for specific domains. Addressing the recall problem would thus require the provision of additional dictionaries and thesauri. Itemization is able to derive the relation type between complex concepts, where the previous strategies invariably fail. However, itemization depends much on the Compound and WordNet strategy. In very complex concept names deriving the correct relation type is rather difficult, so both precision and recall are rather limited. Finally, the Structure Strategy is useful if all other strategies fail. It is able to slightly increase the recall and keep up the precision.

6 Conclusion and Outlook

We presented a new approach for semantic ontology matching that applies an enrichment step to extend correspondences determined with standard match approaches. We exploit linguistic knowledge in new ways to determine the semantic type of correspondences such as is-a and part-of relationships between ontologies. Knowing the intricacies and inconsistencies of natural languages, our approach delivered astonishingly good results in the three real benchmark scenarios. Even in the (German) scenario where background knowledge was practically of no help, we got a recall close to 50 %, and a considerably higher precision. We observed that our rather simple methods mostly achieve already a medium recall and good precision.

Our approach is largely generic and can deal with ontologies from different domains and even with different languages. The enrichment approach can reuse existing match tools, which is both an advantage but also a problem deserving further attention. Standard match tools only aim at finding equivalence correspondences so that many weaker correspondences may not be derivable from the initial match result. To reduce the problem we can use relaxed configuration settings for the initial matching or apply further enrichment strategies utilizing additional information from the ontology (as we have already started with the Structure strategy). We also plan to use additional, domain-specific background sources for improved effectiveness.

Furthermore, we intend to investigate how linguistic methods can be exploited to detect initially falsely detected correspondences, e.g., by taking antonyms or disproved compounds into account. Although this step will not add semantics to the mapping, it is potentially able to increase its precision.

7 Acknowledgment

This study was partly funded by the European Commission through Project "LinkedDesign" (No. 284613 FoF-ICT-2011.7.4).

References

1. Aumueller, D., Do, H.H., Massmann, S.,Rahm, E: Schema and ontology matching with COMA++. Proc. SIGMOD conf. (2005)
2. Bellahsene., Z., Bonifati, A., Rahm, E. (eds.): Schema Matching and Mapping, Springer (2011)
3. Bisetto, A., Scalise, S.: Classification of Compounds. University of Bologna, 2009. In: The Oxford Handbook of Compounding, Oxford University Press, pp. 49-82.
4. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: An Algorithm and an Implementation of Semantic Matching. Proceedings of the European Semantic Web Symposium (2004), LNCS 3053, pp. 61–75
5. Giunchiglia, F., Autayeu, A., Pane, J.: S-Match: an open source framework for matching lightweight ontologies. In: Semantic Web, vol. 3-3 (2012), pp. 307-317
6. van Hage, W. R., Katrenko, S., Schreiber, G. A Method to Combine Linguistic Ontology-Mapping Techniques. In: International Semantic Web Conference (2005), LNCS 3729, pp. 732–744
7. Hamdi, F., Safar, B., Niraula, N. B., Reynaud, C.: TaxoMap alignment and refinement modules: Results for OAEI 2010. Proceedings of the ISWC Workshop (2010), pp. 212–219
8. Jimnez-Ruiz, E., Grau, B. C.: LogMap: Logic-Based and Scalable Ontology Matching. In: International Semantic Web Conference (2011), LNCS 7031, pp. 273–288
9. Massmann, S., Raunich, S., Aumueller, D., Arnold, P., Rahm, E. Evolution of the COMA Match System. Proc. Sixth Intern. Workshop on Ontology Matching (2011)
10. Plag, I., Braun, M., Lappe, S., Schramm, M.: Introduction to English Linguistics, 2nd Revised Edition, Mounton de Gruyter, 2007
11. Raunich, S.,Rahm, E.: ATOM: Automatic Target-driven Ontology Merging. Proc. Int. Conf. on Data Engineering (2011)
12. Reynaud, C., Safar, B.: Exploiting WordNet as Background Knowledge. Proc. Intern. ISWC07 Ontology Matching (OM-07) Workshop
13. Sabou, M., d'Aquin, M., Motta, E.: Using the semantic web as background knowledge for ontology mapping. Proc. 1st Intern. Workshop on Ontology Matching (2006).
14. Shvaiko, P., Euzenat, J.: A Survey of Schema-based Matching Approaches. J. Data Semantics IV (2005), pp. 146–171
15. Spiliopoulos, V., Vouros, G., Karkaletsis, V: On the discovery of subsumption relations for the alignment of ontologies. Web Semantics: Science, Services and Agents on the World Wide Web 8 (2010), pp. 69-88
16. Zhang, X., Zhong, Q., Li, J., Xie, G., Li, H: Rimom Results for OAEI 2008. Proc. 3rd Intern. Workshop on Ontology Matching OM-2008, pp. 182–189
17. Java API for WordNet Searching (JAWS), `http://lyle.smu.edu/~tspell/jaws/index.html`
18. WordNet - A lexical database for English, `http://wordnet.princeton.edu/wordnet/`