Enriching Ontology Mappings with Semantic Relations

Patrick Arnold, Erhard Rahm

University of Leipzig, Germany

Abstract

There is a large number of tools to match or align corresponding concepts between ontologies. Most tools are restricted to equality correspondences, although many concepts may be related differently, e.g. according to an is-a or part-of relationship. Supporting such additional semantic correspondences can greatly improve the expressiveness of ontology mappings and their usefulness for tasks such as ontology merging and ontology evolution. We present a new approach called STROMA (**S**emanT**i**c **R**efinement of **O**ntology **MA**ppings) to determine semantic ontology mappings. In contrast to previous approaches, it follows a so-called enrichment strategy that refines the mappings determined with a state-of-the-art match tool. The enrichment strategy employs several techniques including the use of background knowledge and linguistic approaches to identify the additional kinds of correspondences. We evaluate the approach in detail using several real-life benchmark tests. A comparison with different tools for semantic ontology matching confirms the viability of the proposed enrichment strategy.

Keywords: ontologies, metadata, ontology matching, mapping enrichment, relation type detection, background knowledge

1. Introduction

Ontology matching has been the focus of a large amount of research that led to a broad range of techniques to discover the corresponding or matching concepts between ontologies [26], [8], [2]. Match techniques include lexicographic, structural and instance-based approaches as well as the use of background knowledge and previously found match results. Typically, two related ontologies are matched with each other. The output of the match process is an ontology mapping consisting of the correspondences between matching ontology concepts. Ontology mappings are useful for ontology evolution and different information integration purposes, e.g., for ontology merging.

Email addresses: `arnold@informatik.uni-leipzig.de` (Patrick Arnold), `rahm@informatik.uni-leipzig.de` (Erhard Rahm)

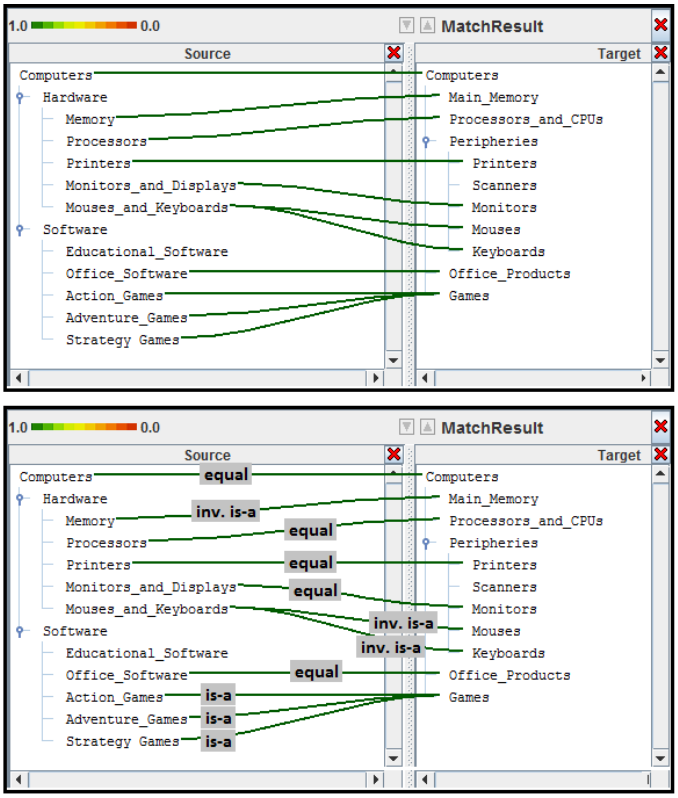

Figure 1: Example of input (top) and intended results (bottom).

A restriction of most match tools, such as COMA++ [1], AgreementMaker [4] or Falcon [14], is that they focus on finding truly matching or equivalent pairs of concepts. However, it would be of great value to determine more expressive mappings including further kinds of correspondences, such as is-a or part-of relations between concepts. Such semantic mappings have been shown to substantially improve ontology merging [21] (see Section 3.2) and to be helpful for ontology evolution [11]. The existing approaches have even difficulties with finding truly equivalent concepts, since similarity-based match approaches are inherently approximative, e.g., if one assumes a match when 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) relation. The importance of semantic ontology mappings has also been recognized by the Ontology Alignment Evaluation Initiative (OAEI),[1] an initiative for evaluating match tools. They provided a full track for detecting both equivalence and subsumption correspondences in

[1] http://oaei.ontologymatching.org/

2011, but had to cancel this track because of insufficient participation [7].

To illustrate the results of current match tools, we show in Figure 1 (top) the result for matching two simple ontologies with the community edition of the state-of-the-art match tool COMA 3.0. Each line represents a correspondence between two concepts. The example shows that not all such correspondences represent equality relations, e.g., *Action_Games – Games*. The figure below illustrates the intended output of our approach with semantically enriched correspondences.

We present a new approach called STROMA (**S**emanT**ic R**efinement of **O**ntology **MA**ppings) to determine more expressive ontology mappings supporting different kinds of correspondences, such as equality, is-a and part-of relations 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 state-of-the-art match tool to determine an initial ontology mapping with approximate equality correspondences. We then apply different techniques (including linguistic approaches and the use of dictionaries) to determine for each correspondence its most likely kind of relationship. In Figure 1 (bottom) we illustrate how the enrichment approach can improve the mapping by identifying several is-a and inverse is-a relations. 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.

Our contributions are the following:

- We propose the use of a two-step enrichment approach to determine semantic ontology mappings that enhances existing match tools.
- We propose the combination of several techniques to determine semantic relations, including the use of new linguistic methods. The approaches can determine equality, is-a (subsumption), inverse is-a, part-of and inverse part-of relations.
- We propose new measures to evaluate the quality of semantic ontology matching for different relations and considering the specifics of the two-phase match approach.
- We provide a comprehensive evaluation of our approach using real-life ontologies. We also provide a comparative evaluation of the new enrichment strategy with two previous tools for semantic ontology matching.

The rest of the paper is organized as follows: We discuss related work in Section 2 and semantic relation types and their application in Section 3. In

Section 4, we explain the basics of our approach and the general workflow that is performed, before we turn towards the strategies used to specify the semantic relation types of correspondences in Section 5. We evaluate our approach for different real-life test cases and compare it with related matching tools in Section 6, and finally conclude in Section 7.

2. Related Work

Only a few tools and studies already try to determine different kinds of correspondences or relationships for ontology matching. S-Match [10], [9] 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 consider S-Match in our comparative evaluation in Section 6.

Taxomap [13] is a match 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 [16]. 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 the evaluation in [25], Taxomap achieved for a mapping scenario with 162 correspondences only a low recall of 23 % and a good precision of 89 % (in later evaluations [25], the relation types were not regarded as the used mappings only comprised equal correspondences). We consider TaxoMap in our evaluation (Section 6).

Aroma [5] is able to detect equivalence and subsumption correspondences. It is an instance-based approach that applies association rules on term sets of concept instances to derive equal and is-a relations. ASMOV [15] exploits lexicographic, structural and instance-based techniques as well as background knowledge (WordNet). They distinguish between equivalence, subsumption and disjoint correspondences, but the authors do not provide an evaluation for these specific types.

In Table 1, we summarize the main features of these tools as well as the STROMA approach to be described. All previous tools are 1-step, i.e., they aim at identifying the different relation types directly when evaluating pairs of concepts. All tools aim at identifying equal and is-a correspondences and a few can also determine "related" types. Finding part-of relations has not yet been supported but will be possible with STROMA. All tools except Aroma exploit background knowledge such as WordNet and apply linguistic techniques to determine correspondences. In addition, structure-based, logic-based, probabilistic and instance-based strategies are being used for semantic matching. To the best of our knowledge, no previous approach has been evaluated with

	S-Match	TaxoMap	Aroma	ASMOV	STROMA
Architecture	1-step	1-step	1-step	1-step	2-step
Supported types	equal, is-a, related	equal, is-a, related	equal, is-a. disjoint	equal, is-a	equal, is-a, part-of, related
Background sources	WordNet	WordNet		WordNet	WordNet, UMLS, OpenThesaurus
Primary techniques	linguistic	linguistic	probabilistic, instance-based	linguistic, structural, instance-based	linguistic, structural

Table 1: Comparison of semantic match tools.

respect to its effectiveness to identify relation types different than equality. Our comparative evaluation for STROMA will also address this important aspect.

Several further studies deal with the identification of semantic correspondence types without providing a complete tool or framework. In [27] reasoning and machine learning are exploited to determine the relation type of a correspondence, where several structural patterns between ontologies are used as training data. An approach utilizing current search engines is introduced in [12]. 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 relation. However, this approach poses different problems, such as the general restrictions of search engine APIs w.r.t. the number of requests sent within a specific time, as well as the rather long execution time for a single element comparison. The approach in [24] uses the Swoogle search engine to detect correspondences and relation types (equal, subset or mismatch) between concepts of many crawled ontologies.

In [16], the authors use so-called Super Word Sets (SWSs) to find correspondences between concepts A, B. This approach utilizes the synsets defined in WordNet, where a synset is a set of synonymous concepts, such as $\{car, auto, automobile\}$. Synsets are linked with each other, for instance the "car" synset may be linked to the "vehicle" synset, where the link type would be hyponym. Given a concept A, and a WordNet synset S in which A appears, the Super Word Set of A, $SWS(A)$, is the aggregation of all hypernyms, hyponyms, meronyms and holonyms of S (for more information on linguistic relations see Section 3.1). The SWS similarity is the number of synsets that $SWS(A)$ and $SWS(B)$ share. They propose a threshold β (20 %) to accept (A, B) as a correspondence and a threshold γ (80 %) to consider it an equivalence correspondence (otherwise, subsumption is assumed). W.r.t. to relation type detection, this is a very vague approach, though, because the number of overlapping synsets does not offer any semantic reason for a true subsumption relation. The quality of the relation type specification has not been evaluated.

Determining the semantic relation type of correspondences is closely related to the problem of identifying complex (one-to-many or many-to-many) correspondences [23]. For example, a many-to-one situation where several concepts of the first ontology are related to the same concept of the second ontology can indicate is-a or part-of relationships between these concepts. Complex correspondences are especially relevant for schema mappings supporting the transformation of data instances from a first schema format to another one, for example to express that two schema elements such as *first_name* and *last_name* should be concatenated to obtain a *name* element [28]. A few approaches have tried to find complex mappings for schema matching, e.g., by analyzing instance data as proposed in [6]. In principle, complex correspondences might also be detected within an enrichment step as proposed for COMA in [17].

3. Semantic Relation Types

We first introduce main linguistic relations that are commonly supported in dictionaries such as WordNet. These relations are closely related to the semantic relation types we want to identify. We further illustrate the value of semantic mappings for ontology merging.

3.1. Linguistic and Semantic Relations

We first characterize the different linguistic relations between words and show some examples in the second column of Table 2 [19]. 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.

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.

For ontology matching, there is no common terminology for semantic relation types. Depending on the chosen perspective and background of the authors we find linguistic terms as well as set-oriented, object-oriented (e.g., UML) or more vernacular terminology for different correspondence types. Table 2 summarizes the main terms for these different perspectives. In this study, we consider all six relation types of the last column (except "mismatch") and use the respective and frequently used terms "equal", "is-a", "inverse is-a", "part-of", "has-a" and "related".

We observed that previous studies (including S-Match and TaxoMap) mostly ignore the relation types "part-of" and "has-a", but only consider type "subsumption" for both is-a and part-of relations. This is obviously too imprecise (e.g., *Student is part of a University* makes more sense than *Student is a kind of University*) so that we aim at determining all mentioned relation types.

Linguistic relation	**Example**	**Set theory (mathematical)**	**UML/OOP term**	**Correspondence type**
Synonymy	river, stream	Equivalence		equal, same as
Antonymy	valley, mountain	Disjoint		mismatch
Hyponymy	apple, fruit	Subset-of	Specialization, Subsumption	is-a, more-specific, less-general
Hypernymy	vehicle, car	Inclusion, Superset-of	Generalization	inverse is-a, less-specific, more-general
Meronymy	roof, building		Aggregation, Composition	part-of, belongs-to
Holonymy	body, leg			has-a
Cohyponymy	oak, maple, birch		(Association)	(related-to)

Table 2: Typical linguistic and semantic relations.

3.2. Using Semantic Ontology Mappings

To illustrate the value of semantic ontology mappings, we discuss their use for integrating or merging ontologies. The typical approach is to first match the ontologies to identify corresponding concepts that need to be represented only once in the merged ontology [20]. All different concepts are also included in the merge result to preserve the information of the input ontologies.

As a simple example we consider the two beverage taxonomies in Fig. 2 together with an equivalence-based match mapping as determined by a standard match tool. Merging these taxonomies would combine the two root concepts as well as the matching concepts *Red Wine* and *Wine*. The merge result would thus not differentiate between these two similar, but not truly equivalent concepts.

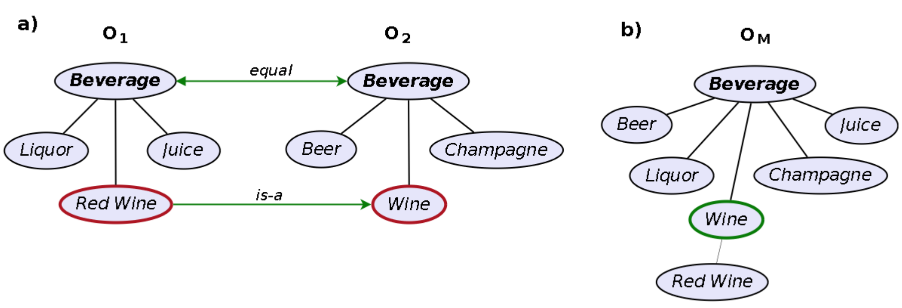

Figure 2: Two ontologies to be merged (2a) and the perfect merge result (2b).

A better merge result becomes possible by not only considering equivalence correspondences but also the other semantic relations within an enhanced match mapping. ATOM is one recently developed merge approach to exploit such semantic input mappings [21, 22]. With such an approach, we can consider that the correspondence between *Red Wine* and *Wine* is of type "is-a" and we would obtain the merge result shown in Fig. 2. This result preserves and correctly places all concepts of the input taxonomies without introducing any redundancy.

4. Overview of Semantic Mapping Enrichment Approach STROMA

An ontology O consists of a set of concepts C and relation 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 relation or correspondence type, and an optional confidence value between 0 and 1 expressing the computed likelihood of the correspondence.

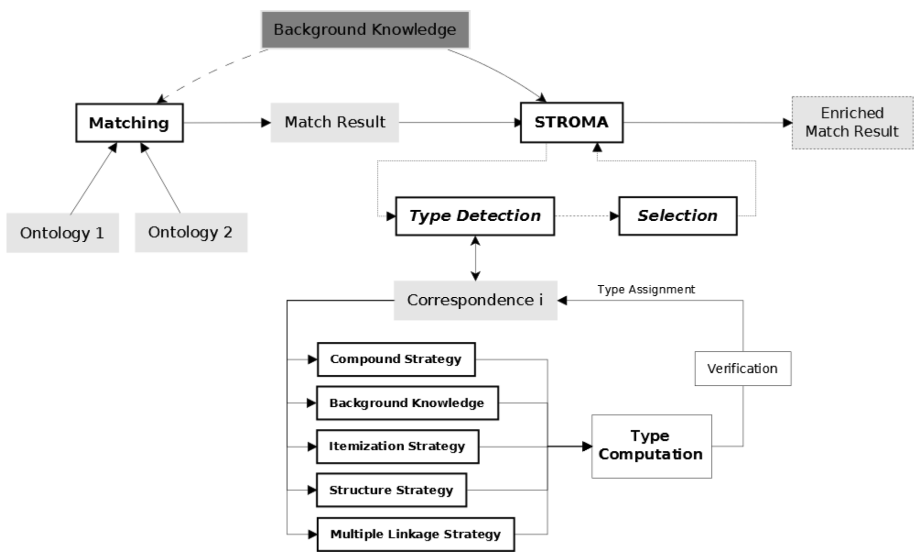

Figure 3: Basic Workflow for Mapping Enrichment.

The basic workflow of our enrichment approach is shown in Figure 3. It consists of two steps: (initial) matching and semantic enrichment with our tool STROMA. The initial matching is performed using a state-of-the art tool for ontology matching such as COMA or AgreementMaker. It takes two ontologies and possibly additional background knowledge sources (depending on the tool) as input and computes an initial match result (a set of correspondences). This match result together with background knowledge sources is the input for STROMA consisting of the substeps of relation type detection and selection. For relation type detection, we currently apply five strategies (Compound, Background Knowledge, Itemization, Structure, Multiple Linkage) for each input correspondence to determine its semantic relation type. The five strategies will be described in the next section. The subsequent selection substep reconsiders the equivalence correspondences from the initial match result for which no relation type could be found.

The proposed two-step approach for semantic ontology matching offers significant advantages. First of all, it is simpler compared to 1-step approaches that try to directly determine the correspondence type when comparing concepts in O_1 with concepts in O_2. The search space for direct matching is the Cartesian product of all pairs of O_1 and O_2 concepts making it difficult to achieve a good match quality and efficiency for large ontologies. In fact, previous approaches for semantic matching could not yet demonstrate their suitability to deal with large ontologies. By contrast, enrichment has to evaluate a relatively compact set of correspondences which is likely to incur relatively small processing requirements. Secondly, STROMA 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, 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. In our evaluation, we will use the COMA match tool that has already shown its effectiveness in many domains [17]. Furthermore, we try to fine-tune the initial match process by using relaxed configurations providing many potential correspondences to find semantic relations weaker than equality. In our final selection step we can filter unlikely equivalence correspondences for which the enrichment step did not find any semantic relation type.

For relation type detection, STROMA needs to combine the results of the five individual strategies. Each strategy returns exactly one type per correspondence or "undecided" if no type can be confirmed. Internally, the results are stored in a matrix. To derive a common result we apply a simple scoring using a pre-determined weight w per strategy. We use weight 1 for most strategies, but assign lower weights to strategies that have been less reliable in our experiments. Table 3 illustrates this with the matrix for a sample correspondence, namely (vehicle, car). The two strategies Compound and Background Knowledge decided correctly on relation type "inverse is-a", while Multiple Linkage decided on "is-a". Itemization and Structure could not decide on any type and returned "undecided" (note that undecided is not a relation type and thus does not have any score). By summing up the individual scores we derive the highest score for "inverse is-a" so that this type is assigned to the correspondence.

There are two issues that may occur:

1. All strategies return "undecided".
2. Two or more types have the same maximal score.

If all strategies return undecided, we apply the equal type to the correspondence, because it is the default correspondence type from the initial ontology matching step. If two types have the same maximal score, we use the following

Strategy	w	eq.	is-a	inv. is-a	has-a	part of	rel.	und.
Compound	1.0	0	0	1	0	0	0	
Background Kn.	1.0	0	0	1	0	0	0	
Itemization	1.0	0	0	0	0	0	0	X
Structure	0.8	0	0	0	0	0	0	X
Multiple L.	0.5	0	0.5	0	0	0	0	
Sum	0.0	0	0.5	**2**	0	0	0	

Table 3: Sample matrix of the internal type calculation.

priority order: equal, is-a / inv. is-a, has-a / part-of, related. However, this case occurred in hardly any of our tests and evaluations. Once the type has been determined, we go into a relation type verification step where we consider additional aspects to confirm or reject the previously made decision (see Section 5.6).

In the final selection step of our workflow we aim at eliminating unlikely equivalence correspondences from the initial match mapping for which no semantic relation type could be found in the enrichment step. The selection step has been introduced to permit the use of relaxed configurations in the initial matching phase retaining correspondences with a low similarity as candidates for non-equivalence correspondences. Given a default threshold θ for the minimal confidence to accept a correspondence in the initial match phase, we now use a weaker threshold θ_w with $\theta_w < \theta$ for the matching step. Consequently, this leads to more correspondences in the initial match result and a generally increased recall at the expense of a lower precision. Given a correspondence C with a confidence value s, we now apply the following selection rules:

1. If $s \geq \theta$ we accept the correspondence to the final mapping without restrictions.
2. If $s \geq \theta_w, s < \theta$ we accept this correspondence only if at least one strategy could determine any relation type. Otherwise we drop the correspondence to improve precision.

With this selection step we aim at an improved recall for the final semantic mapping without much influencing the precision, because each correspondence accepted from the critical interval $[\theta_w, \theta]$ is justified from one of our implemented strategies.

5. Implemented Strategies

In this section, we describe the five implemented strategies to determine the correspondence type as well as the verification step. Table 4 gives an overview of the strategies and the relation types they are able to detect. It can be seen that the Background Knowledge approach is especially valuable as it can help finding all relation types. All strategies are able to identify is-a correspondences.

Strategy	equal	is-a	part-of	related
Compounding		X		
Background Knowledge	X	X	X	X
Itemization	X	X		
Structure		X	X	
Multiple Linkage	X	X		

Table 4: Supported correspondence types per strategy.

From the five approaches, the linguistic techniques Compounding and Background Knowledge are most versatile and generally applicable. By contrast, Structure and Itemization are more specific and also build on the aforementioned strategies. Multiple Linkage does not depend on other strategies, but rather on the relative size of the input ontologies.

The five strategies are based on existing techniques such as string similarity measures and the use of background knowledge but we refine these approaches to identify semantic relation types. Itemization is a new strategy we developed on the basis of our experiments with different taxonomies containing itemizations.

5.1. 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 relation. 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. From a morphological point of view, compounds can be open (as in *high school*), hyphened (as in *bus-driver*) and closed (as in *blackboard*).

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 relation 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 (bottom).

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. The value of 3 is motivated by the observation that English nouns or adjectives consist of at least 3 letters. 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*. We can also eliminate some prefix-derivations, which are generally not semantically related, such as *retail* is a *tail* or *inconsistency* is a *consistency*, although we cannot cover all cases, such as derivations like (disadvantage, advantage).

For this reason, we also keep a list of common English prefixes that indicate derivations (like disadvantage, inconsistency, reaction, unconsciousness etc.). Prefix-derivations generally change the semantics of the stem considerably and even express antonyms in many cases. We therefore discard such correspondences from the match result, since they are probably incorrect.

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 would prevent false conclusions where the pseudo modifier has a length of 3 or more and is no prefix, such as "*nausea* is a *sea*". We found that this approach does not improve our results, because there are some exceptions where the modifier does not match any official word, although it is an endocentric compound. This refers to words that changed their spelling as it is in *holiday* (holy + day) as well as so-called cranberry morphemes, where the morpheme (resp. supposed modifier) occurs only once in a language and has (today) no specific meaning (examples include ***cran****berry*, ***cob****webs* and ***luke****warm*). These morphemes do not represent any words and can thus not be found in word lists or dictionaries. Finally, we discovered technical and scientific compounds whose modifiers are official words of a language, but are too specific to be contained by a standard word list or dictionary, e.g., in the medical domain. For this reason, checking the modifier is rather inconvenient and we did not further pursue this idea.

5.2. Background Knowledge Strategy

In linguistics, the so-called arbitrariness of language is well known indicating that there is frequently no obvious relation between a word and its meaning. For example, the three words *chair*, *stool* and *seat* more or less refer to the same real-world object, a piece of furniture to sit on, but nothing in the spelling or pronunciation of the words indicates this reference, nor do the words resemble each other in any way. On the contrary, the words *cable*, *gable*, *lable*, *maple*, *stable*, *table* etc. are both similar in spelling and pronunciation, but have nothing in common and refer to different concepts.

For this reason, lexical techniques (based on string similarity) as they are often used in match tools, have only limited possibilities to correctly match concept names even for ontologies from the same domain with many similar concepts. These approaches thus miss relevant correspondences, like *(chair, seat)*, but can derive false correspondences, like *(stable, table)*. The use of Background Knowledge sources helps to handle both problems. They may specify that *chair* and *seat* are synonyms despite their low lexical similarity and that the lexically similar terms *stable* and *table* are in no way related. Furthermore, background knowledge can specify specific relation types to be determined for semantic matching.

Our background knowledge strategy can accommodate any data source or dictionary, as long as it can be transformed into a simple list of triples ($term_1$, $relation$, $term_2$). For performance reasons, we use preprocessed and locally stored sources. Currently we use WordNet as well as several other sources for background knowledge (see below).

In case that an open compound C matches a single word W, where W is found in a background dictionary such as WordNet, yet C is not, we gradually remove the modifiers of C in order to detect the relations. We start with the left-hand modifier, remove it and check whether the new form C' is found in the dictionary. If C' is in the dictionary, we check the relation between C' and W, yet have to regard that the actual concept C is a hyponym of C' (as it is a compound). If C' is not contained in the dictionary, we proceed till we reach the head of C. This method does not work with other compound types, though most specific English compounds are always open compounds. We call this technique *Gradual Modifier Removal*.

The implementation of Gradual Modifier Removal for an open compound $C = mH$ consisting of a modifier m and a head H (keep in mind that H can also be a compound) that matches the non-compound concept W is as follows:

1. Is W contained in the dictionary?
   YES: Proceed with step 2.
   No: Type cannot be determined.
2. From C remove the modifier m so that $C' = H$. Is C' contained in the dictionary?
   YES: Proceed with step 3.
   NO: Proceed with step 2.
3. Is $rel(C', W) \in \{equal, isa\}$?
   YES: Return relation type "is-a".
   NO: Type cannot be determined.

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, we could return the correct type (is-a). Since "US Vice President" is a "Vice President" (Compound approach) and "Vice President" is a "Person" based on the dictionary, we assign the is-a relation to the correspondence.

Our background approach exploits the transitivity of synonyms, hypernyms and hyponyms in order to augment its effectiveness. Given a correspondence (C_1, C_2) let's assume that we want to test this correspondence for an is-a relation. We first check whether both concepts are represented in the dictionary. If so, we retrieve all hypernyms of C_1, denoted as $H(C_1)$, and check whether C_2 is contained by this set. If not, we retrieve the hypernyms $H(C_i)$ for each $C_i \in H(C_1)$ and check whether we find C_2 in any of those sets. We continue this procedure until we find C_2 in a hypernym set or reach the root concepts of WordNet.

Currently, **WordNet** is our main source for background knowledge. It contains nouns, verbs, adjectives and adverbs, but we focus on the use of nouns since concept names are typically nouns. We currently use WordNet 3.0 consisting of 117,659 synsets (82,115 noun synsets), 155,287 different words (117,798 nouns) and 206,941 semantic relations (146,312 relations between noun synsets)[2], which were carefully assigned by linguists. We observed that WordNet is a very reliable source that helps finding many non-equal relations that cannot be detected by lexical techniques. On the other hand, WordNet covers only a small part of the English language, which is believed to comprise about a million words, not even including very specific fields such as the chemical or biological domain. Furthermore, the current version of WordNet is from 2006 and does not contain modern terms like *smartphone*, *tablet pc* or *cloud computing*.

Additionally, we use OpenThesaurus and parts of the UMLS Metathesaurus as background knowledge. **OpenThesaurus**[3] is a collaborative open source thesaurus for the German language, which is useful for the German-language mapping scenario of our evaluation (see Section 6). As of October 2013, OpenThesaurus comprises about 102,795 different words in 27,610 different synsets. This makes it less extensive than WordNet, but we observed a massive growth for this source due to an intensive collaboration of volunteers. The unrestricted collaboration seems to introduce some quality issues, but we still consider OpenThesaurus as a reliable source to deal with German-language scenarios.

UMLS is a huge collection of dictionaries and thesauri for the medical and biological domain, containing about 150 different data sources.[4] In our approach, we decided to use the UMLS-Metathesaurus, which is among the largest and, for our purposes, most relevant data source. The thesaurus contains about 157,000 concepts and 141,000 semantic links, which is a remarkable size for the rather specific domain it covers.

5.3. 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

[2]http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
[3]http://www.openthesaurus.de
[4]http://www.nlm.nih.gov/research/umls/

a specific form of cars, but something more general. Hence, there is in general an inverse is-a relation 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 (bottom) 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 relations 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 relation in this case.

5.4. 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 relation between a father concept X of Y and Z (or vice versa). For an is-a relation 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 relation 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 relation 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 relation for the correspondence.

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

5.5. Multiple Linkage

Multiple Linkage is a specific strategy that draws conclusions of schema elements participating in more than one correspondence. If, for instance, a source node s is involved in three matches and thus linked to target nodes t_1, t_2, t_3, it seems that s is more general than either of t_1, t_2, t_3. We would then say that these are inverse is-a relations. Analogously, if a single target node t is linked to several source nodes s, we might consider all these correspondences is-a relations. This strategy is only useful if there are some (1:n) or (n:1) correspondences (complex matches) in a mapping, which often occurs if the source and target schema are of quite different size.

The Multiple Linkage strategy is motivated by some examples where elements like *kitchen chair*, *armchair*, *bar stool* are all linked to a single concept *chair*, which then appears to be a more general concept. However, this approach is rather fragile since false matches in the mapping can lead to wrong conclusions. For instance, if t_1 and t_2 are falsely linked to s and only (s, t_3) is a correct match, there is no reason anymore to justify this as an inverse is-a relation. Therefore, this strategy depends on a high precision of the initial match result and has a lower weight (see Table 3).

5.6. Verification Step

After the relation type of a correspondence is determined based on the five strategies and the combination approach described in Section 4, we perform a verification step where we consider some more subtle aspects to ultimately confirm or revoke the previously made decision. We observed that the identification of is-a 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 Knowledge 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 relation 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").

6. Evaluation

In this section we present a comprehensive evaluation of the proposed STROMA approach for semantic ontology matching and provide a comparative evaluation for S-Match and TaxoMap. We decided on these tools as they are both able to calculate semantic mappings and are freely available. Both tools had sometimes difficulties to import or process our ontologies. S-Match is unable to load (OWL) ontology files so that we had to elaborately convert them to text files. TaxoMap encountered parsing errors in our ontologies, which we could solve by changing the OWL format. We use four scenarios and six match tasks for which we manually determined the presumed perfect match result with semantic correspondence types.

Determining the perfect mapping is much more difficult than for classic match techniques, because in many cases the true relation cannot be unequivocally determined. For example, the correspondence type for (street, road) could be considered of type is-a (as suggested by WordNet) or equal. There might even be different relations 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 for retail stores it might be correct, since a customer would expect *strawberries* to be listed under the concept *berries*.

Even when we manually determine a perfect or near-perfect match mapping or benchmark, we want to determine the overall match quality as well as the match quality for different relation types. Furthermore, evaluating the enrichment approach is complicated by its dependency on the match result of step 1 that might be incomplete and partially wrong. We will therefore introduce different evaluation measures to deal with these issues.

In the remains of this section, we first introduce our measures to evaluate the match quality (Section 6.1) and the used test cases (Section 6.2). We then evaluate the enrichment approach first under best-case conditions based on the (untyped) benchmark mapping (Section 6.3) and second on a match mapping determined by the COMA 3.0 tool (Section 6.4). Next, we evaluate the quality of the individual strategies (Section 6.5) and show how enrichment and selection change the quality of the original COMA 3.0 mapping (Section 6.6). Finally, we compare our approach with S-Match and TaxoMap (Section 6.7).

6.1. Evaluation Measures

We denote the untyped match mapping used as input for enrichment with M and the enriched mapping with semantic correspondences as ME. The presumed perfect mapping or benchmark without and with relation types is denoted with B and BE, respectively. One-step semantic match tools directly determine ME, while the enrichment approach depends on input mapping M. Within the mappings ME and BE we can distinguish disjoint subsets per relation type, e.g., the sets of is-a correspondences, part-of correspondences etc.

Fig. 4 illustrates the different mappings and their possible overlaps. The two overlapping, lengthy ellipses are ME and BE subsets for a specific relation type (e.g. is-a); the left one (not highlighted) for the match result M and the right one (highlighted in dark gray) for the benchmark. Hence, their intersection α contains the correctly typed correspondences for the considered relation type.

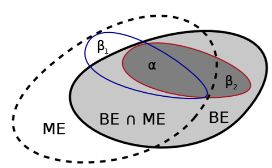

Figure 4: Overlapping correspondences between mapping ME and benchmark BE.

For the untyped match mapping we determine the recall r and precision p measures as well as the combined F-measure f as usual based on the mapping cardinalities (number of correspondences):

$$r = \frac{|B \cap M|}{|B|}, \quad p = \frac{|B \cap M|}{|M|}, \quad f = \frac{2pr}{p + r} \tag{1}$$

In the following we focus on variations for precision and recall; their F-measure combination is always possible according to equation (1). For evaluating the enriched mappings we differentiate two kinds of measures named strict recall / precision (r_s / p_s) and effective recall / precision (r_e / p_e) defined as follows:

$$r_s = \frac{|BE \cap ME|}{|BE|}, \quad p_s = \frac{|BE \cap ME|}{|ME|} \tag{2}$$

$$r_e = p_e = \frac{|BE \cap ME|}{|B \cap M|} \tag{3}$$

The strict measures consider the whole mappings ME and BE as useful for 1-step semantic match approaches. For the enrichment approach, however, these measures are problematic due to the dependency on the input mapping M which may be incomplete and may contain false correspondences (it always holds that $r \geq r_s$ since $|BE| = |B|$ and $|B \cap M| \geq |BE \cap ME|$; $p \geq p_s$ holds at least for $|ME| = |M|$). In particular, the precision p_s suffers from typed correspondences in ME that do not belong to B and thus not to BE, although these correspondences might be correctly typed (e.g., set β_1 in Fig. 4). The recall r_s is reduced by B-correspondences already missing in the input mapping M and thus in ME, e.g., set β_2 in Fig. 4. By contrast, the effective measures focus on the correspondences that are included in the benchmark mapping and determine the fraction of correctly typed correspondences and thus the overall accuracy of relation type detection. It holds that $r_e \geq r_s$ because $|BE| = |B| \geq |B \cap M|$; $p_e \geq p_s$ holds at least for $|ME| = |M|$.

In addition to these overall measures, we evaluate the type-specific match quality by the following measures for strict and effective recall/precision (*type* may refer to any relation type such as is-a):

$$r_{s(type)} = \frac{|BE_{type} \cap ME_{type}|}{|BE_{type}|}, \quad p_{s(type)} = \frac{|BE_{type} \cap ME_{type}|}{|ME_{type}|} \tag{4}$$

$$r_{e(type)} = \frac{|ME_{type} \cap BE_{type}|}{|ME \cap BE_{type}|} \quad p_{e(type)} = \frac{|ME_{type} \cap BE_{type}|}{|ME_{type} \cap BE|} \tag{5}$$

We illustrate the introduced measures with an example. Let's assume that $|B| = |BE| = 100$, $|M| = |ME| = 150$ and $|B \cap M| = 80$. This results in a recall $r = 80/100 = 0.8$ and a precision $p = 80/150 = 0.53$ for input mapping M. Let's say that $|BE \cap ME| = 70$, meaning that 70 of the 80 correspondences in $|B \cap M|$ were correctly typed. We have $r_e = p_e = 70/80 = 0.88$. The strict recall is $r_s = 70/100 = 0.7$ and the strict precision $p_s = 70/150 = 0.47$.

Assume now that the number of is-a correspondences in the benchmark is $|BE_{isa}| = 80$ and in the match result $|ME_{isa}| = 100$. Let's also assume that the number of correctly typed is-a correspondences is $|BE_{isa} \cap ME_{isa}| = 45$ and that $|BE_{isa} \cap ME| = 55$, meaning that 55 of the is-a correspondences in the benchmark also occur in the mapping, with 45 of them being correctly denoted as "is-a". Furthermore, we assume $|ME_{isa} \cap BE| = 60$, i.e., 60 of the is-a correspondences in ME are also in the benchmark, albeit only 45 of them are truly of type "is-a". We thus have a strict recall $r_{s(isa)} = 45/80 = 0.56$ and an effective recall $r_{e(isa)} = 45/55 = 0.81$. The strict precision is $p_{s(isa)} = 45/100 = 0.45$, while the effective precision is $p_{e(isa)} = 45/60 = 0.75$.

No	Domain	Lang.	#Corr.	equal	is-a	part-of	related
B_1	Web Directories	DE	340	278	52	5	5
B_2	Diseases	EN	395	354	40	1	0
B_3	TM Taxonomies	EN	762	70	692	0	0
B_4	Furniture						
1)	*Amazon-Ebay*	EN	136	15	108	10	0
2)	*Wikipedia-Ebay*	EN	87	3	83	0	1
3)	*Wikip.-Amazon*	EN	138	16	115	7	0

Table 5: Overview of evaluation scenarios and benchmark mappings

6.2. Evaluation Scenarios

For our evaluation, we used six ontology matching tasks from four scenarios of different domains. For each match task we manually defined the presumably perfect benchmark mapping with relation types. Table 5 provides key information about the match tasks and benchmark mappings, 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/part-of/related).

The first scenario (benchmark B_1) matches the Yahoo (797 concepts) and Google Web taxonomies (product catalogs of shopping platforms, 2,223 concepts). The ontologies are in German language, so WordNet is of no use here. This scenario contains many itemizations, which the other scenarios lack. The second match task (B_2) addresses the medical domain and uses an extract of 395 correspondences between the diseases catalogs of Yahoo (5,436 concepts) and dmoz (10,084 concepts). B_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 originally consists of about 1,600 correspondences [18], about half of which with 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 B_3, leaving us with 762 correspondences of type equal, is-a and inverse is-a.

The fourth scenario includes three match tasks involving three ontologies; these tasks and benchmark mappings have been newly developed by us for this study. We recognized that the previous benchmarks are rather inconvenient to compare our approach with other tools. Being a German-language scenario, tools using WordNet have difficulties to deal with B_1, while B_2 is only an extract from a larger benchmark, making it difficult to compare it with the mappings produced by other tools calculating the full mapping. Finally, we do not possess the schemas for B_3, but only the mapping. Since all other tools need the schemas for matching, this standard was inapplicable, too. We therefore created a new benchmark based on the furniture category of amazon.com (174

	r	p	f
B_1	.48	.70	.57
B_2	.66	.90	.77
B_3	.87	.92	.90
$B_{4.1}$	.64	.99	.77
$B_{4.2}$	.37	.97	.53
$B_{4.3}$	.35	.80	.49

(a) Non-equal types

	r	p	f
B_1	.96	.89	.92
B_2	.99	.96	.97
B_3	.90	.57	.70
$B_{4.1}$	.93	.24	.38
$B_{4.2}$	1	.05	.10
$B_{4.3}$	1	.19	.32

(b) Equal-types

	r	p	f
B_1	.87	.87	.87
B_2	.96	.96	.96
B_3	.87	.87	.87
$B_{4.1}$	.67	.67	.67
$B_{4.2}$	.39	.39	.39
$B_{4.3}$	.43	.43	.43

(c) Overall result

Table 6: Evaluation against benchmark

concepts), ebay.com (25 concepts) and Wikipedia[5] (184 concepts). We consider the associated match tasks as relevant, as they cover a challenging real-world scenario with many non-equal correspondences. The three ontologies roughly cover the same domain, with Amazon and Wikipedia being more specific and Ebay being more general.

6.3. Evaluation for Perfect Input Mapping

We first evaluate the STROMA enrichment approach using the untyped benchmark mappings as input. This represents a best-case scenario ($M = B$) where the task is only to determine the correct relation type per correspondence and strict recall/precision become identical to effective recall/precision.

Table 6 shows the achieved recall / precision and F-measure results for the six match tasks. Table 6a) 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 F-measure between 49 and 90 %, indicating a good effectiveness. Precision was especially good (70 to 99%), while recall was somewhat limited, especially in scenarios $B_{4.2}$ and $B_{4.3}$ that turned out to be quite challenging.

Table 6b) only considers the equal-type. In the first and second scenario, the equal relation dominates (about 90 % of all correspondences) and in these cases both recall and precision are very high. By contrast, in the third and fourth scenario we achieve only a poor precision. 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 and fourth scenario. We observe that the results for non-equal correspondences in 6a) and for equal correspondences in 6b) are inversely interrelated.

Finally, Table 6c) summarizes the overall results considering all correspondence types. F-measure values varies significantly; the first three test cases are well solved (F-measure of 87-96%) while the fourth scenario is problematic as we will further discuss in the comparison with other tools. In the overall re-

[5]http://en.wikipedia.org/wiki/Category:Furniture

sults, recall and precision (and thus F-measure) are always equal as we deal with effective recall and precision in this experiment.

6.4. Evaluation for COMA Input Mapping

In the second set of experiments we apply the ontology matching tool COMA 3.0 [17] for the initial matching to determine a real, imperfect input mapping for the enrichment step. We use the default workflow of COMA (All Context Workflow), as it proved to be a reliable strategy for rather general scenarios as we use in this evaluation. This strategy compares the concept names, the parent names, the full paths as well as the location of the concept nodes in the ontology structure. The techniques used include string matchers like TriGram and TF/IDF (more details are given in [17]).

Since we had no ontologies for B_3, we could not generate a mapping for this scenario. We here focus on scenarios 1 and 2 and handle scenario B_4 in Section 6.7 when we compare our approach to other match tools.

	r	p	f
B_1	.71	.74	.72
B_2	.67	.54	.60

(a) Quality of initial match result

	r_e	p_e	f_e	r_s	p_s	f_s
B_1	.50	.53	.51	.14	.21	.17
B_2	1	.94	.97	.38	.47	.42

(b) Results for non-equal types

	r_e	p_e	f_e	r_s	p_s	f_s
B_1	.97	.96	.97	.78	.76	.77
B_2	.99	1	.99	.71	.55	.62

(c) Results for equal type

	r_e	p_e	f_e	r_s	p_s	f_s
B_1	.94	.94	.94	.66	.69	.67
B_2	1	1	1	.67	.54	.60

(d) Overall results

Table 7: Evaluation with COMA 3.0 match results

Table 7 shows the results for the COMA-based experiments. Table 7a) 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).

Table 7b) shows the recall and precision for the detected non-equal types, where we consider both the effective and strict recall resp. precision. We knew that r_s of b) must be lower than r in Table 6a), because in the initial match result some typed correspondences were missing. Still, the strict recall for B_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.

Table 7c) shows the results for the equal correspondences and eventually Table 7d) shows the overall results for all kinds of correspondences. Since most correspondences are of the equal type in B_1 and B_2, most correspondences were correctly typed, and therefore r_s in 7d) is only slightly below the result in 7a).

Considering the effective recall and precision, the results reveal that we could type 94 % resp. 100 % of the correspondences in the match result correctly. This shows that the relation type detection works very precisely if we disregard false or missing correspondences.

6.5. Effectiveness of STROMA Strategies

We also ran our test cases with each of the five STROMA strategies for relation type detection to identify their individual strengths and weaknesses.

Compounding achieves a good precision and practically works in all domains, even for non-English languages (Germanic Languages). Its recall is mostly limited because of the different possibilities of how an is-a relation can be expressed. All strategies (except Multiple Linkage) use the compound strategy, so that we consider it the most significant strategy of our approach.

Background Knowledge proved to be a very precise approach allowing a precision close to 100 %. Table 8 shows how background knowledge improves the quality of the relation type detection for our test scenarios. Background knowledge thus helped in most cases, particularly for $B_3 - B_4$ where non-equal correspondences dominate. The low improvement of mapping quality in B_2 and the absence of any improvement in B_1 is not a weakness of the background knowledge approach, but a consequence of the overall good results determined by the other strategies in these scenarios. For example, OpenThesaurus was able to assign several relation types correctly in B_1, but these types were already supported by other strategies (e.g. Itemization), so that OpenThesaurus did not provide any further information for the mapping. In B_2, UMLS had also difficulties to retrieve additional knowledge, as the initial F-measure of 94 % does not allow much space for improvement.

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. Still, we had many itemizations in B_1 and B_4 where this strategy was valuable.

The **Structure Strategy** had only little impact on the recall and is only an additional technique if the other strategies fail.

Multiple Linkage can increase the recall notably if the match result does not contain too many false matches and if there are many complex matches. Otherwise, this strategy has no effect on the mapping.

The verification step increased our precision considerably. We obtained a boost of about 10 % in precision in the highly hierarchical ontologies of B_1 and B_2. In the rather flat ontology scenarios the verification step had no impact, though.

Generally, the strategies are very effective in detecting is-a relations and confirming equal relations, while they reach their limits w.r.t. the part-of and related type. For instance, none of the 10 part-of correspondences in $B_{4.1}$, e.g., (Dining Chairs, Dining Sets) or (Bookcase Ladders, Bookcases), was found.

	B_1	B_2	B_3	$B_{4.1}$	$B_{4.2}$	$B_{4.3}$
with BK	.87	.96	.87	.67	.39	.43
without BK	.87	.94	.82	.65	.31	.38

Table 8: Effective F-measure with and without using background knowledge.

Although some of these correspondences could be found by a strategy similar to compounding based on matching modifiers (as in the bookcase example), such a technique leads to poor precision and fails in many cases (e.g., a nightstand is not part of night, or an apple tree is not part of an apple). Therefore, background knowledge appears to be the most promising strategy for such correspondence types, although dictionaries can only cover a relatively small share of all reasonable part-of relations.

6.6. Influence of Enrichment and Selection on the Initial Mapping

We now analyze how the match quality of the COMA input mapping is affected by our enrichment and selection step. The default COMA selection filter is 0.4, i.e., correspondences must meet this minimal similarity threshold. For semantic enrichment we lowered this threshold to 0.2 to obtain more input candidates for finding non-equal correspondences. In the selection step, we only accept the correspondences between 0.4 and 0.2 for which a relation type could be found or confirmed. The question is whether this approach can improve standard match quality (ignoring relation types) compared to having no enrichment phase.

Table 9 shows the achieved match quality results for using only COMA with threshold 0.4 (Table 9a) and threshold 0.2 (Table 9b). Table 9c) shows the match quality for COMA threshold 0.2 in combination with STROMA enrichment and selection. Comparing the two COMA-only cases we observe the expected behavior that for the lower threshold recall is improved while precision and F-measure decrease for all test scenarios. Comparing the two cases with COMA threshold 0.2 (Table 9b vs. 9c), we observe that the additional enrichment and selection improves precision and F-measure in all test cases, especially for scenarios $B_1 - B_3$.

Comparing the results in Table 9c) with Table 9a) we observe that enrichment and selection also improves F-measure compared to the COMA default match strategy in five of six cases. This is due to a considerable boost in recall because of the reduced threshold and the applied linguistic and background knowledge strategies finding non-equal correspondences using our approach. For B_1 and B_2 there are only few non-equal correspondences so that recall could only slightly improve while the reduced precision leads to an almost unchanged or even decreased F-measure. By contrast, scenario B_4 comprises many non-equal relations so that our approach could significantly improve F-measure.

Still the results for scenario B_4 remain at a low level, especially for $B_{4.2}$ and $B_{4.3}$. We found out that in many cases our solution detects a correct relation type between two concepts, but that this link is not the most specific one and is

	r	p	f
B_1	.71	.74	.72
B_2	.67	.54	.60
$B_{4.1}$	.57	.55	.56
$B_{4.2}$	.08	.64	.14
$B_{4.3}$	.07	.38	.12

(a) COMA standard match quality ($\theta = 0.4$, no enrichment).

	r	p	f
B_1	.78	.51	.62
B_2	.70	.53	.60
$B_{4.1}$	.83	.18	.30
$B_{4.2}$	.39	.07	.12
$B_{4.3}$	.34	.06	.11

(b) COMA match quality with reduced thresholds ($\theta = 0.2$) and without enrichment.

	r	p	f
B_1	.77	.56	.65
B_2	.68	.56	.61
$B_{4.1}$	.73	.56	.63
$B_{4.2}$	.31	.38	.34
$B_{4.3}$	.18	.24	.21

(c) Match quality for COMA (reduced threshold ($\theta = 0.2$) + STROMA for semantic enrichment.

Table 9

thus not in the benchmark. For instance, COMA discovered the correspondence ("Courting Bench", Kids Furniture.Benches"), which is erroneous as a courting bench is not primarily suited for kids rooms. Our approach correctly discovered the is-a relation (Courting bench is a bench) but this correspondence is not part of the benchmark mapping. The benchmark contains the correspondence ("Courting Bench", "Other furniture.Benches") which was not discovered by COMA.

6.7. Comparison of Match Tools

We now compare STROMA with the two previous semantic match tools S-Match and TaxoMap. Again, the STROMA results are based on the mappings generated with COMA 3.0. For the evaluation, we use the full furniture benchmark (scenario B_4). The results for the three match tasks are shown in Tables 10 – 12. The tables are organized as in Section 6.4: a) shows the general match quality where the relation type is disregarded, b) and c) show the effective and strict match (relation type detection) quality regarding non-equal resp. equal correspondences, and d) shows the overall semantic match quality. As we already observed, the scenarios $B_{4.2}$ and $B_{4.3}$ are especially challenging. They involve the Wikipedia taxonomy with many specific concepts not contained in other dictionaries (such as *bean bag*, which is a specific type of chair, or *cassone*, which is a specific type of chest). For these two scenarios, TaxoMap was not

	r	p	f
Stroma	.73	.56	.63
S-Match	.74	.24	.37
TaxoMap	.21	.62	.31

(a) Quality of matching

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.78	.80	.79	.62	.48	.54
S-Match	.95	.85	.90	.70	.20	.31
TaxoMap	.12	.54	.20	.12	.54	.20

(b) Results for non-eq. types

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.69	.56	.62	.60	.41	.49
S-Match	.15	1	.27	.13	.67	.21
TaxoMap	.87	.76	.81	.87	.76	.81

(c) Results for equal type

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.77	.77	.77	.62	.47	.53
S-Match	.85	.85	.85	.63	.21	.31
TaxoMap	1	1	1	.21	.62	.31

(d) Overall results

Table 10: Amazon-Ebay Scenario ($B_{4.1}$)

	r	p	f
Stroma	.31	.38	.34
S-Match	.10	.03	.04

(a) Quality of matching

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.78	.88	.82	.23	.31	.26
S-Match	1	1	1	.07	.02	.03

(b) Results for non-eq. types

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	1	.50	.66	.67	.22	.33
S-Match	1	1	1	1	.43	.60

(c) Results for equal type

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.85	.85	.85	.24	.30	.27
S-Match	1	1	1	.10	.03	.04

(d) Overall results

Table 11: Wikipedia-Ebay Scenario ($B_{4.2}$)

	r	p	f
Stroma	.18	.24	.21
S-Match	.40	.10	.15

(a) Quality of matching

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.75	.75	.75	.09	.13	.11
S-Match	.77	.57	.65	.37	.03	.06

(b) Results for non-eq. types

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.90	.90	.90	.63	.48	.54
S-Match	.13	1	.22	0.02	.67	.04

(c) Results for equal type

	r_e	p_e	f_e	r_s	p_s	f_s
Stroma	.77	.77	.77	.15	.20	.17
S-Match	.58	.58	.58	.14	.03	.05

(d) Overall results

Table 12: Wikipedia-Amazon Scenario ($B_{4.3}$)

able to return any correspondence so that we provide only results for the two other tools.

We ran S-Match (April 2011 Build) and TaxoMap (Version 3.5) in their standard configuration, and COMA 3.0 in relaxed settings as illustrated before. We did not change any configurations during the three experiments. Since TaxoMap allows manual threshold configuration, we tried different combinations to retrieve more correspondences, but constantly obtained the same result as with the standard configuration.

Regarding the untyped match results (tables (a)), we observe that COMA 3.0 provides the best F-Measure in all scenarios. S-Match suffers from a generally very low precision especially for the second and third scenario (10% or less). TaxoMap obtained the best precision in the first scenario, but only a very low recall. It seems to use too restrictive selection thresholds. In the second and third scenario the tool was too restrictive to return any correspondence.

With regard to the overall quality of relation type detection (tables (d)), we find that the proposed enrichment approach based on COMA mappings outperforms TaxoMap and S-Match in all three match tasks for the strict F-measure. This is because of the quality of the enrichment strategy and because of the good quality of the initial COMA mapping. The overall quality is mainly determined by non-equal correspondences which dominate in the considered test scenarios. As can be seen in the tables (b), the enrichment approach (C) achieves always better strong F-measure results for non-equal correspondences than S-Match (S) and TaxoMap (T).

Regarding the effective measures, S-Match achieves somewhat better results than the enrichment approach for the two first scenarios (effective F-measure of 0.85/1 vs 0.79/0.85). However, the effective measures are not well suited for comparing different tools as they determine the degree of correctly typed correspondences in $BE \cap ME$ which differs substantially between tools. In fact, S-Match and TaxoMap mostly have substantially fewer correspondences in $BE \cap ME$ than our enrichment strategy. This facilitates the correct relation detection especially if only simple correspondences are considered. We found that TaxoMap only finds simple correspondences (with perfect precision). For the second scenario with its 87 correspondences, S-Match found only 9 rather simple correspondences such as (Table, Tables). Hence, we believe the comparison between different tools should be based on the strong measures while the effective measures are useful for tool-specific evaluations, e.g., to compare different configurations and test cases.

We also analyzed the execution times of STROMA and the match tools. We did not optimize STROMA for performance but could run all six benchmark tests within 2 - 18 seconds (on a commodity PC) favored by the relative small size of the input mappings determined by COMA. For the three small B_4 match tasks, S-Match, TaxoMap and COMA also needed only a few seconds so that there were no significant differences in runtime. For the larger match task B_1, COMA+STROMA needed 52+11=63 seconds while S-Match and TaxoMap needed 280 and 74 seconds, respectively. For larger match tasks, the runtime for using STROMA is thus mainly determined by the match tool used in the inital step while enrichment is relatively fast depending on the size of the input mapping.

7. Conclusion and Outlook

We presented a new approach called STROMA for semantic ontology matching that applies an enrichment step to extend correspondences determined with standard match approaches. We exploit linguistic techniques and background knowledge in new ways to determine the semantic type of correspondences such as is-a and part-of relations between ontologies. Knowing the intricacies and inconsistencies of natural languages, our approach delivered astonishingly good results in the considered benchmark scenarios and outperformed previously developed semantic match tools.

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 potential problem. Most 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 used relaxed configurations and include justified correspondences in the match result.

The proposed linguistic strategies turned out to be effective and useful within further strategies. To improve recall and thus match quality, the use of background knowledge is especially valuable. Hence, we aim at using additional background knowledge to further improve STROMA. In particular, we will gather background knowledge from web sources such as Wikipedia and Wiktionary that provide definitions (including semantic relations) for many words. Linguistic resources such as BabelNet and UBY are also promising candidates for additional background knowledge. We will further consider alternate tools for the initial matching and extend the proposed techniques.

8. References

[1] D. Aumueller, S. Do, H. H.; Massmann, and E. Rahm. Schema and ontology matching with coma++. In *SIGMOD*, 2005.

[2] Z. Bellahsene., A. Bonifati, and E. Rahm. *Schema Matching and Mapping*. Springer, 2011.

[3] A. Bisetto and S. Scalise. Classification of compounds. In *The Oxford Handbook of Compounding*, pages pp. 49–82. Oxford University Press, 2009.

[4] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Agreementmaker: Efficient matching for large real-world schemas and ontologies. *Proc. VLDB Endow.*, 2(2):1586–1589, August 2009.

[5] Jérôme David, Fabrice Guillet, and Henri Briand. Association rule ontology matching approach. *International Journal on Semantic Web and Information Systems*, pages 27–49, 2007.

[6] Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Halevy, and Pedro Domingos. imap: Discovering complex semantic matches between database schemas. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 383–394, New York, NY, USA, 2004. ACM.

[7] Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2011. In *OM*, 2011.

[8] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2013.

[9] Fausto Giunchiglia, Aliaksandr Autayeu, and Juan Pane. S-match: An open source framework for matching lightweight ontologies. *Semantic Web*, 3(3):307–317, 2012.

[10] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-Match: An Algorithm and an Implementation of Semantic Matching. In Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *Proceedings of the 1st European Semantic Web Symposium*, volume 3053 of *Lecture Notes in Computer Science*, pages 61–75. Springer, May 2004.

[11] Anika Groß, Júlio Cesar dos Reis, Michael Hartung, Cédric Pruski, and Erhard Rahm. Semi-automatic adaptation of mappings between life science ontologies. In *Proc. 9th DILS (Data Integration in the Life Sciences) Conf.*, pages 90–104, 2013.

[12] Willem Robert Van Hage, Sophia Katrenko, and Guus Schreiber. A method to combine linguistic ontology-mapping techniques. In *In International Semantic Web Conference*, pages 732–744, 2005.

[13] Fayal Hamdi, Brigitte Safar, Nobal B. Niraula, and Chantal Reynaud. Taxomap alignment and refinement modules: results for oaei 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel F. Cruz, editors, *OM*, volume 689 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[14] Wei Hu and Yuzhong Qu. Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):237 – 239, 2008.

[15] Yves R. Jean-Mary and Mansur R. Kabuka. Asmov: Ontology alignment with semantic validation. In *Joint SWDB-ODBIS Workshop, Vienna, Austria*, 2007.

[16] Jung Ae Kwak and Hwan-Seung Yong. Ontology matching based on hypernym, hyponym, holonym, and meronym sets in wordnet. In *International journal of Web & Semantic Technology (IJWesT)*, volume 1, 2010.

[17] S. Massmann, S. Raunich, D. Aumueller, P. Arnold, and E. Rahm. Evolution of the COMA match system. In *Proc. Sixth Intern. Workshop on Ontology Matching*, 2011.

[18] Eric Peukert and Katja Pfeifer. Mapping text mining taxonomies. In *6th International Conference on Knowledge and Information Retrieval*, 2013.

[19] I. Plag, M. Braun, S. Lappe, and M. Schramm. *Introduction to English Linguistics*. Mounton de Gruyter, 2007.

[20] Rachel Pottinger. Mapping-based merging of schemas. In *Schema Matching and Mapping*, pages 223–249. Springer-Verlag, 2011.

[21] Salvatore Raunich and Erhard Rahm. ATOM: Automatic target-driven ontology merging. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 1276–1279, Washington, DC, USA, 2011. IEEE Computer Society.

[22] Salvatore Raunich and Erhard Rahm. Target-driven merging of taxonomies with ATOM. *Information Systems*, 2013.

[23] Dominique Ritze, Johanna Völker, Christian Meilicke, and Ondrej Sváb-Zamazal. Linguistic analysis for complex ontology matching. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel F. Cruz, editors, *OM*, volume 689 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[24] Marta Sabou and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *In Proc. of the Int. Workshop on Ontology Matching (OM-2006)*, 2006.

[25] Brigitte Safar. Exploiting wordnet as background knowledge. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, *Proceedings of the Workshop on Ontology Matching (OM2007) at ISWC/ASWC2007, Busan, South Korea*, November 2007.

[26] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics IV*, 3730:146–171, 2005.

[27] Vassilis Spiliopoulos, Vouros George A., and Vangelis Karkaletsis. On the discovery of subsumption relations for the alignment of ontologies. *Web Semant.*, 8(1):69–88, March 2010.

[28] Brian Walshe, Rob Brennan, and Declan O'Sullivan. A comparison of complex correspondence detection techniques. In Pavel Shvaiko, Jérôme Euzenat, Anastasios Kementsietsidis, Ming Mao, Natasha Fridman Noy, and Heiner Stuckenschmidt, editors, *OM*, volume 946 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.